AD-A090 682 RAND CORP SANTA MONICA CA F/G 1/5

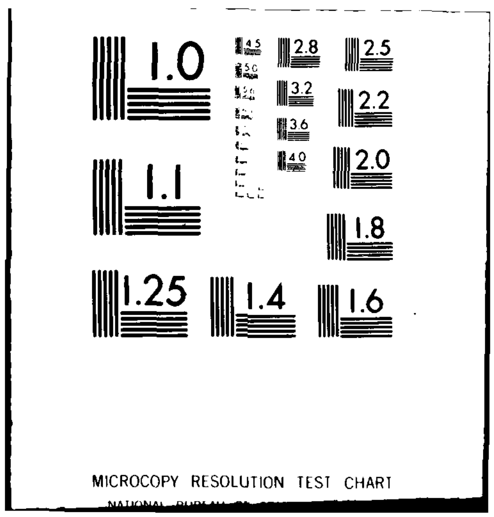
TSARINA: USER'S GUIDE TO A COMPUTER MODEL FOR DAMAGE ASSESSMENT--ETC(U)

JUL 80 D EMERSON F49620-77-C-0023

UNCLASSIFIED RAND/N-1460-AF NL

1 of 2

AD A
090682


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
MICROCOPY RESOLUTION TEST CHART

A RAND NOTE

TSARINA--USER'S GUIDE TO A COMPUTER MODEL FOR DAMAGE ASSESSMENT OF COMPLEX AIRBASE TARGETS

Donald Emerson

July 1980

N-1460-AF

Prepared For

The United States Air Force

Rand
SANTA MONICA, CA. 90406

The research reported here was sponsored by the Directorate of Operational Requirements, Deputy Chief of Staff/Research, Development, and Acquisition, Hq USAF, under Contract F49620-77-C-0023. The United States Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

Published by The Rand Corporation

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER N-1460-AF	2. GOVT ACCESSION NO. AD-A090682	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) TSARINA: User's Guide to a Computer Model for Damage Assessment of Complex Airbase Targets		5. TYPE OF REPORT & PERIOD COVERED Interim
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Donald/Emerson		8. CONTRACT OR GRANT NUMBER(s) F49620-77-C-0023
9. PERFORMING ORGANIZATION NAME AND ADDRESS The Rand Corporation 1700 Main Street Santa Monica, CA 90406		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Requirements, Programs & Studies Group (AF/RDQM) Ofc, DCS/R&D & Acquisition Hq USAF, Washington, D.C. 20330		12. REPORT DATE Jul[illegible]980
		13. NUMBER OF PAGES 39
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

No Restrictions

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Models
Military Air Facilities
Attacks
Computer Programs
Damage Assessment
Targets
Vulnerability

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

See Reverse Side

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

RPN 3509

N-1460-AF TSARINA: User's Guide to a Computer Model for Damage Assessment of Complex Airbase Targets. D. Emerson. July 1980.

Description of the TSARINA computer program, developed to examine conventional air attacks against complex targets and to assess losses and damage to categories of resources and to buildings and other facilities. TSARINA permits damage assessments of attacks on an airbase complex composed of up to 500 individual targets (buildings, taxiways, etc.), and 1000 packets of resources. Targets may be grouped into 20 vulnerability categories, and different types of personnel, equipment, munitions, spare parts, and other support resources can be distinguished. TSARINA determines the actual impact points by Monte Carlo procedures and the losses and damage are assessed using "cookie-cutter" weapon-effects approximations. TSARINA may be employed separately as a general-purpose model or used in conjunction with the TSAR (Theatre Simulation of Airbase Resources) computer model to assess the impact of airbase damage on sortie generation capabilities and to evaluate proposals for improving those capabilities at an airbase or set of airbases. Detailed user instructions and a listing of the program are included. 123 pp. (Author)

A RAND NOTE

TSARINA--USER'S GUIDE TO A COMPUTER MODEL FOR DAMAGE ASSESSMENT OF COMPLEX AIRBASE TARGETS

Donald Emerson

July 1980

N-1460-AF

Prepared For The United States Air Force

Rand
SANTA MONICA, CA. 90406

PREFACE

This Note describes TSARINA, a special modification of Rand's Airbase Damage Assessment (AIDA) computer model that has been developed for examining conventional air attacks against complex targets and for assessing losses and damage to various categories of resources as well as to various buildings and other facilities. TSARINA may be employed as a general purpose damage assessment model, or as a special purpose model for use with the TSAR (Theater Simulation of Airbase Resources) computer model. This Note includes detailed user instructions as well as a listing of the program.

TSARINA was developed by Rand for use with the TSAR model for studying means of sustaining and improving wartime sortie generation capabilities, despite unexpected demands and sudden unpredictable resource shortages imposed by air attacks. TSARINA is basically a Monte Carlo computer model that can generate sample patterns of airbase damage for incorporation into the TSAR simulation, or statistical summaries of the damage and resource losses for multiple samples of the specified attack.

TSARINA has a variety of possible applications. It can be used separately to assess the casualties and losses that would be sustained from air attacks on airbases (or other complex targets), and to assess the impact of various dispersal and/or hardening proposals on the expected losses. It can also be used in conjunction with the TSAR simulation model to assess the impact of airbase damage on sortie generation capabilities, and to evaluate proposals for improving those capabilities at an airbase or a set of airbases.

TSARINA has been used in a study for the Air Staff of the personnel casualties and War Reserve Materiel (WRM) losses that might be sustained in high-level conventional conflict in Europe or Korea, and it is currently being applied in conjunction with TSAR to examine alternative proposals for improving the Air Force's wartime sortie generation capabilities, under Project AIR FORCE. The model, which has been discussed with many groups within the Air Force, has been transferred to the Office of the Assistant Chief of Staff, Studies & Analyses. This Note is being published to provide documentation for the model and to introduce it to a wider audience. The computer program is available from The Rand Corporation.

This work was conducted under the Project AIR FORCE research project entitled "Strategies To Improve Sortie Production in a Dynamic Wartime Environment."

SUMMARY

This Note describes a new version of the AIDA airbase damage assessment computer program that has been developed to assess losses to various on-base resources, as well as damage to runways, taxiways, buildings, and other facilities. The model may be used either as a general-purpose, complex-target damage assessment model, or as a special-purpose model in support of the TSAR simulation program. When used with TSAR, multiple trials of a multi-base airbase-attack campaign can be assessed with TSARINA, and, in a continuous computer operation, the impact of those attacks on sortie generation can be derived using the TSAR simulation model.

TSARINA, as presently configured, permits damage assessments of attacks on an airbase (or other) complex that is composed of up to 500 individual targets (buildings, taxiways, etc.), and 1000 packets of resources. The targets may be grouped into 20 different vulnerability categories, and many different types of personnel, equipment, munitions, spare parts, TRAP (tanks, racks, adaptors, and pylons), building materials, and POL (petroleum, oils, and lubricants) can be distinguished. The attacks may involve as many as 50 weapon-delivery passes and 10 types of weapons. Both point-impact weapons (such as general-purpose bombs and precision-guided munitions) and area weapons (such as cluster bomb units (CBUs)) can be accommodated.

TSARINA determines the actual impact points (pattern centroids for CBUs) by Monte Carlo procedures--i.e., by random selections from the appropriate error distributions. Weapons that impact within a specified distance of each target type are classed as hits, and estimates of the damage to the structures and to the various classes of support resources are assessed using "cookie-cutter" weapon-effects approximations. In addition to the weapon-effects procedures used with AIDA, this model also permits use of a novel two-level cookie-cutter representation for assessing damage to the various classes of resources.

For each trial computation of an attack, the program determines the fraction of each target covered by the circular damage coverage patterns, and the results include estimates of the overall damage to each target and to all resource classes that are colocated with that target. In addition, the output includes an estimate of the total damage sustained by each type of resource at its various storage locations. The attack may be repeated automatically for several trials to provide statistics on the average damage levels to each of the targets and to each type of resource.

A maximum of five targets may be designated as runways or taxiways suitable for aircraft operations, and the model will examine these to see if an area of a user-specified size is available for aircraft operations; if not, the minimum number of craters that would need to be repaired to obtain an area of that size is determined.

The TSARINA program is written in FORTRAN IV, and should be readily adaptable to other computer systems, as was the widely used AIDA model. This Note provides a full discussion of the use of TSARINA as a general-purpose damage assessment model, and outlines in detail the special requirements for its use in conjunction with the TSAR simulation program. Most features of the model are illustrated with a sample problem. Appendixes include a description of TSARINA input requirements, definitions of all variables and arrays found in TSARINA common statements, and a listing of the complete TSARINA source code.

GLOSSARY

Resource class	All airbase support resources are grouped into seven classes: personnel, equipment and AGE, aircraft spare parts, munitions, TRAP, building materials, and POL.
Resource type	Different types of resources may be distinguished within each resource class; e.g., different categories of aircraft maintenance specialists. Resource, when used alone, implies a resource type.
Resource packet	A user-specified percentage of a given resource is referred to as a resource packet; resources are located within the target complex as packets.
Target	A target is represented by a rectangle that is located in an X-Y coordinate system; individual buildings, runways, taxiways, parking areas, etc., can be designated as targets.
Target complex	A target complex, such as an airfield or an industrial area, is a collection of rectangular targets.
Target type	A target type is specified for each target; all targets of the same type have the same vulnerability, and all resource types of the same resource class located at the same type of target have the same vulnerability.
GP	General-purpose (bomb).
PGM	Precision-guided munition.
POL	Petroleum, oils, and lubricants.
TRAP	Tanks, racks, adaptors, and pylons.
WRM	War reserve materiel.
AGE	Aerospace ground equipment.

ACKNOWLEDGMENTS

The author would like particularly to acknowledge Felix Kozaczka and Louis Wegner of the Rand staff for a variety of suggestions that should make the TSARINA program, and this documentation, more useful and understandable.

CONTENTS

PREFACE iii

SUMMARY v

GLOSSARY vii

ACKNOWLEDGMENTS ix

Section
I. INTRODUCTION 1

II. TARGET DATA 3

III. DAMAGE ASSESSMENTS FOR TSAR 5
TSAR Relationships Between Facilities and Resources 6
Identification of Airbase Facilities 8
Procedures to Ensure Resource Category Compatability 12

IV. WEAPON EFFECTIVENESS DATA 15
Point-Impact Weapons 15
CBU Munitions 19

V. SAMPLE PROBLEM 20
Input 20
Output 27

Appendix
A. DETAILED DESCRIPTION OF TSARINA INPUT 37
B. DEFINITIONS OF VARIABLES AND ARRAYS USED IN TSARINA "COMMON" STATEMENTS 59
C. TSARINA SOURCE CODE AND COMMENTS 70
D. PROGRAMMERS' NOTES 122

I. INTRODUCTION

TSARINA (TSAR INputs using AIDA) is a modified version of the AIDA (AIrbase Damage Assessment) computer model*; it was developed to generate airbase damage estimates for a campaign of air attacks, and to organize those results for direct entry into the TSAR (Theater Simulation of Airbase Resources) sortie generation model,** which can assess the impact of the destructive effects of attacks. Several key changes have been made to the AIDA model so that the on-base location of resources (e.g., personnel, munitions, aircraft spare parts, etc.) can be readily associated with various targets (structures/facilities), and so that different MAEs (mean areas of effectiveness) and/or Pks (kill probabilities) can be defined for the different resources. These changes also permit a novel two-level "cookie-cutter"*** representation of the effectiveness of weapons against the various classes of resources. In addition, the various effectiveness values may be different for direct hits and for near misses. With these added input data, TSARINA generates estimates of the losses among the various on-base resources, in addition to the estimates of hits and facility damage that are generated by the original AIDA model.

TSARINA may be used either as a special-purpose model in support of the TSAR simulation, or as a general-purpose damage assessment model. When used with TSAR, multiple trials of a multi-base airbase-attack campaign can be evaluated with TSARINA, and, in a continuous computer operation, the impact of

---

*R-1872-PR, AIDA: An Airbase Damage Assessment Model, D. E. Emerson, The Rand Corporation, September 1976.

**R-2584-AF, An Introduction to the TSAR Simulation Program: Model Features and Logic, D. E. Emerson, The Rand Corporation, June 1980.

***See p. 15.

those attacks on sortie generation can then be derived using the TSAR simulation model. When TSARINA is used for damage estimates only, the various protocols required for use with TSAR (Section III) may be ignored.

Since AIDA, and this new version of AIDA, employ identical target and attack representations, the reader is referred to the AIDA report for a discussion of these concepts. The emphasis in this note will be on the special features associated with TSARINA. With both AIDA and TSARINA the user is able to specify the size, location, and nature of several hundred rectangular targets and the characteristics of up to 50 weapon-delivery passes.* Targets can be categorized into 20 vulnerability classes, and up to 10 types of weapons may be employed in any given attack; point-impact and CBU munitions may be used in the same attack. Both TSARINA and AIDA are basically Monte Carlo models; however, an expected-value mode is available as an option for evaluating damage to aircraft shelters.

---

*The maximum numbers of targets and weapon-delivery passes are readily changed.

II. TARGET DATA

In AIDA and TSARINA, the facilities on an airbase are represented as a target complex consisting of a number of rectangular-shaped targets (e.g., runways, parking ramps, buildings); the size, location,* and type of each target are specified. With TSARINA the user may also specify the resources that are associated with each target. The resources to be identified may be grouped into seven categories; personnel, equipment, aircraft spare parts, munitions, TRAP, building materials, and POL. And within each class, different subclasses may be distinguished by type; for example, the personnel class may distinguish pilots, crew chiefs, radar repair specialists, and weapons loaders. Hereafter, the term resource will refer to a particular resource class and type.

The user may specify the percentages of the different resources that are located at each target. Thus, personnel with different specialties may be located at different facilities; AGE and other equipment can be located in various buildings or parked in designated areas; and different kinds of munitions, TRAP, etc., can similarly be located in various proportions in various on-base locations.

The losses estimated for each resource depend upon the attack weapon type, weapon impact location, resource class, and target type, location, and orientation.** In each case, it is assumed that the resource is distributed uniformly

---

*AIDA's restrictions on target location have been relaxed for TSARINA. It is no longer necessary that all targets be located in the first quadrant of the X-Y coordinate system, and the targets may cover an area as large as 32000 x 32000 dimension units. When the target location entries are not entirely within the first quadrant, TSARINA automatically translates the coordinate system to place all targets (and attacks) in the first quadrant for processing. Furthermore, an auxiliary program is available that will convert dimensional information structured for the Eglin AFB MASSIVE computer program into the format required for TSARINA.

**All subcategories of a given resource class, located within targets of the same type, are assumed to have the same vulnerability.

within the target area, and that the aggregate losses for that resource are the sum of the losses estimated to be sustained at each of the target locations. Additional weapon effectiveness data must be supplied to complement these extended target descriptors and the user is given considerable flexibility as to how these data are expressed, as discussed in Section IV.

Any number of resource packets may be associated with each target, except that there may be no more than 1000* packets in total. The designations used to specify the different classes of resources are defined in the subsequent discussion of supplementary TGT cards. The "integer" designators that are to be assigned to each type of resource are selected by the user; the only constraint is that the integers chosen are not greater than the size of the corresponding storage array.

TSARINA's treatment of runways is identical with that described for AIDA. Runways must always be identified as type #1 targets; up to five may be entered. When the minimum clear length and width needed for flight operations are identified, and the "minimum repair requirements" option is requested (on the CONT card), all runway targets are searched to find whether or not an uncratered area of the required size exists, and if not, what the smallest number of crater repairs would be to attain that amount of clear space.

---

*This restraint is easily changed by redefining MXITEM (line 85, page 72) and redimensioning the STOCKS array (line 34, page 71 et al.)

III. DAMAGE ASSESSMENTS FOR TSAR

When TSARINA is to be used to generate damage assessments for he TSAR simulation, it is necessary to make the specifications for the targets and the resources consistent with the conventions used in TSAR. If TSARINA is not to be used with TSAR, but as a general-purpose damage assessment model, the reader may skip to Section IV.

The TSAR computer model is a large, complex, task- oriented event-simulation model that has been developed to interrelate the number of effective sorties that can be generated in wartime at a set of airbases to the resources that are available. It has been designed to provide a means of assessing the potential contributions of various ideas for improving and sustaining sortie capabilities, despite unexpected wartime demands and sudden unpredictable resource shortages imposed by airbase attacks. When used with TSAR, TSARINA generates sample patterns of airbase damage and loss that are incorporated into the simulated TSAR scenario. Special requirements must be satisfied when using TSARINA in order for TSARINA outputs to interface satisfactorily with TSAR.

These additional requirements are a necessary reflection of the considerable complexity and flexibility of the TSAR simulation, and arise as a result of

- o The mechanism employed in TSAR to associate resources with facilities
- o The conventions used in TSAR to identify facilities
- o The procedures provided to permit disparate resource categorizations in the two models.

The following subsections will expand on each of these items.

TSAR RELATIONSHIPS BETWEEN FACILITIES AND RESOURCES

The TSARINA results include estimates of (1) the percentage loss for each resource class at each target and (2) the overall percentage loss for each type of resource at all locations. When used in TSAR, one or the other of these percentages are applied to the quantities of each resource at risk at the time of the attack. For most resources, losses are estimated using the second of these two type of estimates. Since the derivation of this loss percentage in TSARINA is based on a nominal user-specified on-base resource disposition, and is not affected by the particular consumption experience in the TSAR simulation, the use of this overall estimate effectively assumes that the on-base disposition of a given subcategory of resource, expressed in percent, is not dependent upon the absolute level of such resources. The first type of loss estimate is used only for on-duty maintenance personnel, aircraft maintenance equipments, and reparable spare parts; these resources are accounted for individually within TSAR, so that their assigned location at the time of the attack is known.

The TSAR simulation is able to associate these resources with particular facilities because it assigns personnel and equipment to each individual task and those tasks (are normally assumed to) occur in a designated facility. The maintenance personnel engaged in on-equipment maintenance are assumed to be in close proximity to the aircraft to which they are assigned. On-duty maintenance personnel who are not actively engaged in on-equipment maintenance are assumed to be awaiting assignment at their respective work centers, or, in the case of the flightline personnel in a COMO (Combat Oriented Maintenance Organization), in a particular flightline facility. Specialists involved in parts repair jobs are assumed to be in the facility designated for the repair of the particular type of part that they are repairing, as are all reparable spare parts.

Losses among resources engaged in on-equipment maintenance always are conditioned by a separate assessment of the likelihood that the aircraft is exposed to damage (this relationship will be explained shortly). For the others, the nominal TSAR-TSARINA logic presumes that unassigned on-duty personnel and equipment, and personnel and equipment engaged in off-equipment parts repair, are in their designated facilities, and that the loss percentages to these resources are related to the damage to those facilities. In this way, appropriate loss percentages are assigned these mobile resources at the instant of the attack. Damage estimates for facilities and their associated resources are only passed to TSAR for designated targets; this is done by entering the facility number (as defined in the TSAR data structure) on the appropriate TSARINA TGT cards. If the user wishes the target-dependent loss percentages to be transferred to TSAR for some targets, but not for others, the facility numbers for the latter targets should not be entered.

Also, if the user wants the target-dependent loss percentage to be used for some of the resources that are nominally associated with a particular facility, but not for others of those resources, the target-dependent value will be overridden in TSAR for whichever resources have a specific damage estimate transmitted to TSAR. Thus, if on-duty radar technicians have been located in a variety of facilities for the TSARINA airbase attack simulation, the TSARINA estimate of percent damage for those technicians will normally be applied to all such personnel not engaged in on-equipment aircraft maintenance within TSAR, even though a different personnel loss percentage is associated with their normal work facility, and it is passed to TSAR. To prevent the target-dependent estimate from being overridden, the user must either not include the specific locations for these technicians, or specifically instruct TSARINA not to transfer the estimate by use of the EQUI card format, as discussed later in this section.

---

*When the damage to some, but not all, resources is specified, the dimensions in TSAR's subroutine BOMB currently permit at most 50 damage specifications for each resource class.

Careful attention to which targets are identified with a TSAR facility number, and which resources are located specifically in TSARINA, provides the user substantial flexibility for controlling the damage information transferred from TSARINA to TSAR and thereby representing his knowledge of expected on-base conditions in the simulation data base.

IDENTIFICATION OF AIRBASE FACILITIES

Some airbase facilities are handled in an aggregate manner in TSAR--aircraft shelters, runways, taxiways, and aircraft parking ramps--and for these facilities the user simply specifies the target-type number (on the DATA card, see p. 41) that is used in TSARINA to designate these kinds of facilities. Other facilities--those that relate to specific maintenance functions--are identified within TSAR with specific facility numbers, and these facility numbers must be identified in TSARINA if damage to these facilities is to be communicated to TSAR from TSARINA. The following paragraphs discuss the particular data requirements for each of the several target classes.

Runways

The maximum number of facilities that are distinguished in TSAR is designated by the variable NOFAC. By TSAR convention, the runway is identified by that maximum number, i.e., NOFAC.* For TSARINA the value of NOFAC is specified in the source code of the MAIN subroutine, along with the dimensions of other key TSAR arrays. The user should consult the comments in the MAIN routine for a description of the particular TSARINA configuration he has available; if the configuration is at variance with his TSAR configuration, changes to the TSARINA configuration are easily accomplished.

---

*Since TSAR is only concerned with the repairs that are to be accomplished to provide the minimum area for flight operations, without regard to which runway is to be repaired, a single datum is required.

Aircraft Parking Ramps

For aircraft parking ramps, TSARINA generates an estimate for the percentage of exposed aircraft that are damaged by computing the expected percentage of aircraft parts that would be destroyed, if they were dispersed at random in these areas, using the rules given in Section IV for specifying aircraft parts vulnerability. In using this estimate, TSAR first subtracts the number of aircraft that could be sheltered from the number of aircraft on the ground, and assumes that the remainder are unsheltered. The survival of the unsheltered aircraft is then checked by comparing a random number for each aircraft with the exposed aircraft damage percentage estimated in TSARINA.

Taxiways

For taxiways TSARINA simply counts the number of point-impact weapons that hit on, or within the weapon radius of, these targets. The TSARINA output identifies these hits with the TSAR target number for taxiways. Hits on the taxiways are communicated to TSAR as damage to facility number NOFAC-1.

Maintenance Shops

In TSAR each on-equipment task and parts repair job is associated with a particular shop or facility. If the shop is damaged, the parts repair jobs and certain designated on-equipment tasks can not be accomplished. Facilities #1 through #24 are reserved for the locations of the various specific maintenance functions (i.e., work-centers or shops) that are designated by the same number. When the facility number for one of these maintenance functions is entered on the TGT card in TSARINA, damage to that facility is communicated to TSAR.

Other facility numbers are reserved for other functions: #27 relates to aircraft reconfiguration, #28 to munitions loading, #29 to fuel servicing, and #30 to munitions assembly; facilities #31, #32, and #33 are used to define the assembly points for the flightline specialists associated with squadrons one, two, and three, respectively, in a AFR 66-5, or COMO, maintenance organization. Damage to each of these facilities will also be transmitted to TSAR when their facility number is identified on the appropriate TGT card.

When any of these facilities sustain one or more "hits," TSARINA generates an estimate of the percentage of the facility that sustains damage, as well as estimates of the percentage losses sustained by the personnel, AGE, and parts present in the facility at the time of the attack.

Aircraft and Aircraft Shelters

TSAR assesses aircraft damage and loss by drawing a random number for each aircraft on-base at the moment of attack, and by comparing that number with an estimate of the fraction of the aircraft that are damaged and/or killed. In TSARINA a damage fraction is generated both for sheltered and unsheltered aircraft, and TSAR applies the latter fraction to that number of on-base aircraft that exceed the then-current capacity of the shelters, as noted earlier.

Damage to shelters and sheltered aircraft are handled uniquely in TSARINA, in that the damage estimate may be generated either with the Monte Carlo mode (as for other targets) or with the expected-value mode. The Monte Carlo mode is required if the user wishes to consider specific resources that might be stored in the shelters; if he does not have that requirement, he may reduce TSARINA computer processing by use of the expected-value mode.*

With TSARINA the user designates the target type that he has assigned to shelters on the DATA card (see p. 41), and enters the location and size data for each shelter either with the TGT or TGT2 type cards, depending upon whether the Monte Carlo or expected-value mode is to be used for assessing damage.

---

*Since TSAR compares a random number with the damage estimate for each aircraft, losses will vary from trial to trial even though the expected value mode is used. When the Monte Carlo mode is used, the damage estimate used for these comparisons will also vary between trials.

Weapon effectiveness entries are also handled differently for these targets. In all, 14 different weapon effectiveness data may be entered; 13 for each type of weapon, and one that is presumed to apply for all weapons. To enter these data in TSARINA, 14 cards are entered for each point-impact type weapon, rather than the one or two cards that are used in AIDA. The entry on the first card is interpreted as the effective miss distance against buttoned-up shelters; a hit is recorded whenever a weapon strikes within this distance of a shelter. Data entered on the second through ninth cards control the damage estimates for personnel, AGE, spare parts, munitions, TRAP, and building materials that might be stored in the shelters when a weapon strikes within the first effective miss distance. The several resource damage criteria that are available are explained in Appendix A. When the expected value mode is used, only the third through fifth of these cards is used, and the entries are interpreted as the percent losses sustained by personnel, AGE, and spare parts in a shelter that has closed doors when a hit has been recorded.

Four additional data are entered on the tenth, twelfth, thirteenth and fourteenth EMD cards; the first defines the effective miss distance against aircraft in a shelter with an open door, the next two define the probability of damage to an aircraft when the shelter door is closed, and when it is open, and the last defines the probability of kill of the shelter itself, when a weapon strikes within the first effective miss distance. The factor that controls the fraction of the damaged aircraft that are not reparable is entered on the DATA card, as explained on page 41. This factor is presumed to apply for whatever mix of weapons has been used in the attack.

The various assessments of aircraft and shelter damage, and of the losses among resources associated with the aircraft at the moment of the attack, are communicated to TSAR as class 8 and class 10 in a particular facility; the data structure for these transmissions can be inferred from the code in subroutine DAMAGE.

PROCEDURES TO ENSURE RESOURCE CATEGORY COMPATIBILITY

A requirement for the proper functioning of TSAR and TSARINA is that the level of detail at which resources are specified must be the same for both models. Although it might be adequate for some damage assessments for resource types to be lumped into broad categories, such aggregate results would not be useful for the simulation if these resources were treated in more specific terms in TSAR. Thus, one cannot specify "air-to-air missile losses" in the TSARINA output if TSAR requires that losses of AIM-7s and AIM-9s be distinguished.

The required consistency can be achieved in three different ways. First, TSARINA inputs may specify the location data separately for each type of, say, munition that is to be distinguished in TSAR. Or, if the user cannot distinguish the locations for some types of munitions, he may locate them as a group, and then specify that the same damage should be reported to TSAR for each type of munition in the group; the EQUIvalence card-format is used for this purpose. Lastly, if the individual locations for many, or all, of the various types of any class of resources cannot be distinguished, the user may locate these resources and report their common damage to TSAR by the simple expedient of not specifying a resource number. To clarify these options, consider the following table:

Type-Number Specified in TSARINA Location Data	Types Specified in TSAR	User Action Required
1	1 2	EQUI data card
3	3	None
4	4 5 6	EQUI data card
0	7 8 9 10 11 12	None

Assuming that the resource class is TRAP, for example, this is how one would handle the situation in which the TSAR data base distinguished 12 different types of TRAP, but location data could only be specified for four different categories, and only one corresponded to the TSAR data base, e.g., TRAP #3. Location data for TRAP #1 and TRAP #2, for TRAP #4, #5, and #6, and for TRAP #7 through #12 cannot be distinguished, presumably, for this example. The only special action the user needs to take is to define the actual types of TRAP, implied by the TSARINA type numbers 1 and 4, using the EQUI data cards. It should be noted that the numbers need not be in any special sequence and that type #0 always implies "all types not otherwise specified."

When dealing with personnel and munitions, two other special features come into play. The first of these is concerned with the distinction between on-duty and off-duty personnel, which are treated differently within TSAR. To indicate that a particular category of personnel is off-duty, the number 1000 is added to the nominal personnel type when their location data are entered.* When this is done, casualties among on- and off-duty personnel are estimated separately in TSARINA and reported separately to TSAR without any additional user input. Specification of personnel type #1000 implies all off-duty personnel types that are not otherwise specified.

The other special number convention permits the user to distinguish the locations and damage to assembled and unassembled munitions; unassembled munitions are identified simply by adding the number 100 to the nominal munition type designation. Thus, if munition #5 were assembled AIM-7s, munition #105 would refer to unassembled AIM-7s. These special identity numbers for personnel

*This feature, as currently implemented, behaves as described when TSAR is dimensioned for 1 to 100 different personnel types. Defining that dimension as NOPEOP, aircrews are identified for TSAR with the number NOPEOP +1; thus, if NOPEOP = 100, aircrews are personnel type #101.

and munitions may either be assigned to the resources when their location data are specified, or subsequently, using the EQUIvalence card type.

IV. WEAPON EFFECTIVENESS DATA

For each weapon type the user has several options for specifying its effectiveness against the various types of targets (facilities) and against the various resources that may be present at those targets. These options are expressed by up to 17 data for each target type and for each type of weapons; these 17 data are entered using the EMD card (see page 50) and up to 16 supplementary cards for each type of weapon.

As described more fully in the AIDA report, the basic mathematical representation that is used for the effectiveness of a point-impact weapon is what has been called a cookie-cutter--that is, a uniform probability of kill over a circle of specified radii. Integration of actual weapon effects kill probability contours over the many rectangular targets was rejected in order to limit computer processing. Although the cookie-cutter approach to representing weapon effects has been retained in TSARINA, an attempt has been made to provide the user somewhat greater flexibility for these approximate weapon effects representations. As developed below, there are eight different possibilities provided for estimating losses for each of the different classes of resources, including what might be defined as a two-level cookie-cutter; i.e., an inner circle with a specified Pk, and an outer circle defined such that the average Pk is just one-fourth that for the inner circle.

POINT IMPACT WEAPONS

For Each Target Type Except Aircraft Shelters

The first and tenth data for each target type are the mean radius of effectiveness (R1) against the structures of that target type; these data are entered on the first and tenth of the EMD cards. The first entry applies to near misses and the tenth to direct hits; if a radius is not entered for direct hits, the first data applies in both cases. The TSARINA results include the fraction of each target that is covered by circles of these radii, one circle for each weapon delivered; the covered area is interpreted as the percent of target damage. Entries on the second and eleventh EMD cards are also effectivesness radii (R2); the interpretation of these radii is dependent upon the assessment criteria in use. The entries on the third through eighth, and the twelfth through seventeenth, EMD cards control damage assessments of the various resources in all facilities. These assessments, for the various resource classes, depend upon damage factors that are entered in the following locations:

EMD Card		
Near Miss	Direct Hit	Resource Class
Third	12th	Personnel
Fourth	13th	AGE and equipment
Fifth	14th	Spare parts
Sixth	15th	Munitions and POL
Seventh	16th	TRAP
Eighth	17th	Building materials

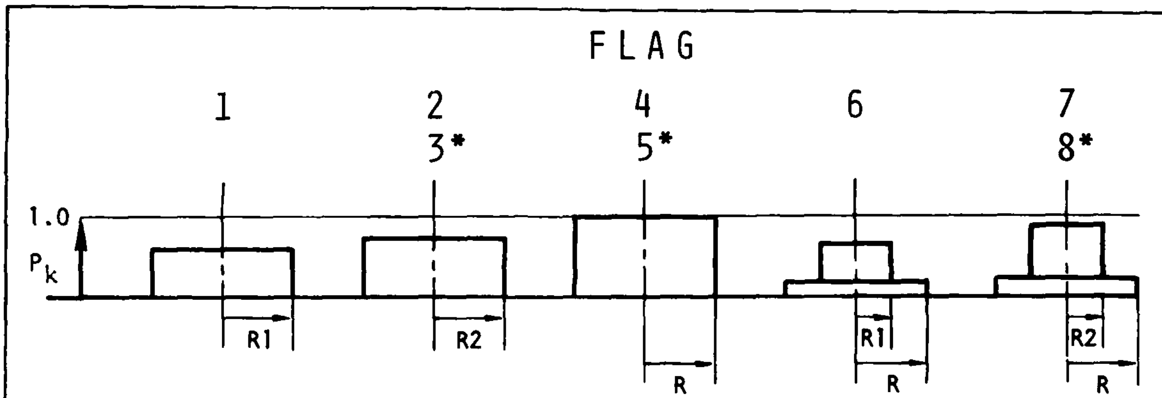
As noted above, the "near miss" values apply when data are not entered for direct hits. If any data are entered for a particular target type on the eleventh through seventeenth cards, only data from those cards will be used in assessing direct hits; i.e., blank entries are interpreted as zero.

The values entered with these 12 resource damage cards are interpreted either (1) as effective kill radii against these resources, (2) as probabilities of kill of such resources within the radii R1 or R2, or (3) as a radius and a kill probability that jointly define a "two-level" cookie-cutter, depending upon the value of the FLAG input described below.

Since munitions and POL would not be expected to be present together, little flexibility is lost with the dual definition used with the sixth (and fifteenth) card; it is important, of course, that the user be clear as to the distinction.

The interpretations of the weapon effectiveness parameters on these twelve cards are specified by the user on the ninth EMD card--i.e., by the value of the variable called "FLAG" in the program.

The user has eight options for handling the vulnerability of each of the six classes of resources, and FLAG communicates these six choices with a 6-digit code. These eight options are illustrated below:



*Effectiveness is assessed for these options only when target is within the radius R1.

The eight weapon effectiveness parameter options that are identified by the 6-digit code are defined as:

0 Ignore this class of resource

1 Value represents probability of kill of these resources within a circle of radius R1

2 Value represents probability of kill of these resources within a circle of radius R2

3 As in 2, given that radius R1 intersects the target perimeter

4 Value is the radius of kill of these resources

5 Value is the radius of kill of these resources, given that radius R1 intersects the target perimeter

The last three values are a combination of a radius and a probability of kill.

6 Value is (1) the radius (times 1000) of an area within which the Pk is one-fourth that value of Pk within R1, plus (2) the value of Pk within R1 (thus 60080, for example, specifies that Pk is 0.80 within R1, and 0.20 in the annular area between R1 and 60 feet).

7 Value is the radius (times 1000) of an area within which the Pk is one-fourth that value of Pk within R2, plus the value of Pk within R2

8 As in 7, given that radius R1 intersects the target perimeter

Thus FLAG = 321475 would imply that the six classes of resources are to be assessed by options 3, 2, 1, 4, 7, and 5, respectively, for the particular weapon type and target type for which it is listed.

Aircraft Shelters

No distinction is made between a direct hit and a near miss for aircraft shelters; the entry on the first EMD card is interpreted as the effective miss distance against aircraft in shelters with closed doors, and the entry on the tenth card is interpreted as the effective miss distance against aircraft in shelters with open doors. When shelters are handled with the Monte Carlo mode the entries on the second through ninth cards are used to assess losses to resources in a closed shelter in a manner consistent with other target types, as just explained. When the user does not identify specific resources within the aircraft shelters, and they are treated with the expected-value mode, the entries on the third through fifth cards are interpreted as the loss probabilities for personnel, AGE, and spare parts when a weapon strikes within the radius specified on the first card. In either case, the probabilities that aircraft are damaged in closed and open shelters, and the probability that the shelter itself is lost, are given by the entries on the twelfth through fourteenth cards, respectively.

CBU MUNITIONS

The entries on the first EMD card for CBU munitions specify the reliability of the dispenser and the length and width of a rectangular pattern of bomblets; the weapon-type number and the number of cards to be entered for these weapons are also specified. The user may enter up to seven supplementary cards with each EMD card for CBU munitions. The first supplementary card is blank, and the entries on the second through seventh supplementary cards are to be interpreted as the percentage of the six resource classes that would be expected to be lost if the bomblet pattern covered their location. Intermediate cards must not be omitted; e.g., if only equipment and TRAP losses were of interest, two blank supplementary cards would still have to precede the third card and two more precede the sixth card.

V. SAMPLE PROBLEM

The layout of the test base is shown in Figure 1. This base consists of a 200 ft x 7000 ft runway, a parallel taxiway, two connecting taxiways, four shops, three aircraft parking ramps, six aircraft shelters, two vehicle parking areas, and four barracks. To examine the effectiveness of a bombing attack against this target complex with TSARINA, one first must describe the target elements and the attack in a common coordinate system. Each target is defined by its westernmost corner, its size, and its orientation;* the attack heading and the desired mean point of impact fix the attacks. For this illustration four aircraft are assumed to attack each of two points on the runway in an effort to "cut" it; in addition three aircraft attack each of the two sets of three shelters, and three aircraft attack the complex of four shops. Each attacker drops either 18 or 24 bombs with a range error probable (REP) and deflected error probable (DEP) of 300 feet and 150 feet, respectively. The intervalometers are set for stick lengths between 1500 and 2500 feet.

Several different types of resources are present in these facilities and are at risk to the attack. Varying quantities of personnel and AGE are in the four shops. In addition, several types of AGE are in the vehicle parking areas. Many off-duty personnel are in the barracks. Several of the aircraft shelters have TRAP and munitions stored internally.

INPUT

Figures 2 and 3 reproduce the card images needed to describe this sample problem and to control the assessment. For clarity, the control (CONT) card, the TSAR data (DATA) card, the target (TGT) cards, and the attack (ATT) cards are listed in order; the

*An auxiliary program is available for converting dimension data prepared for the MASSIVE program (developed at Eglin AFB) into the format required for TSARINA.

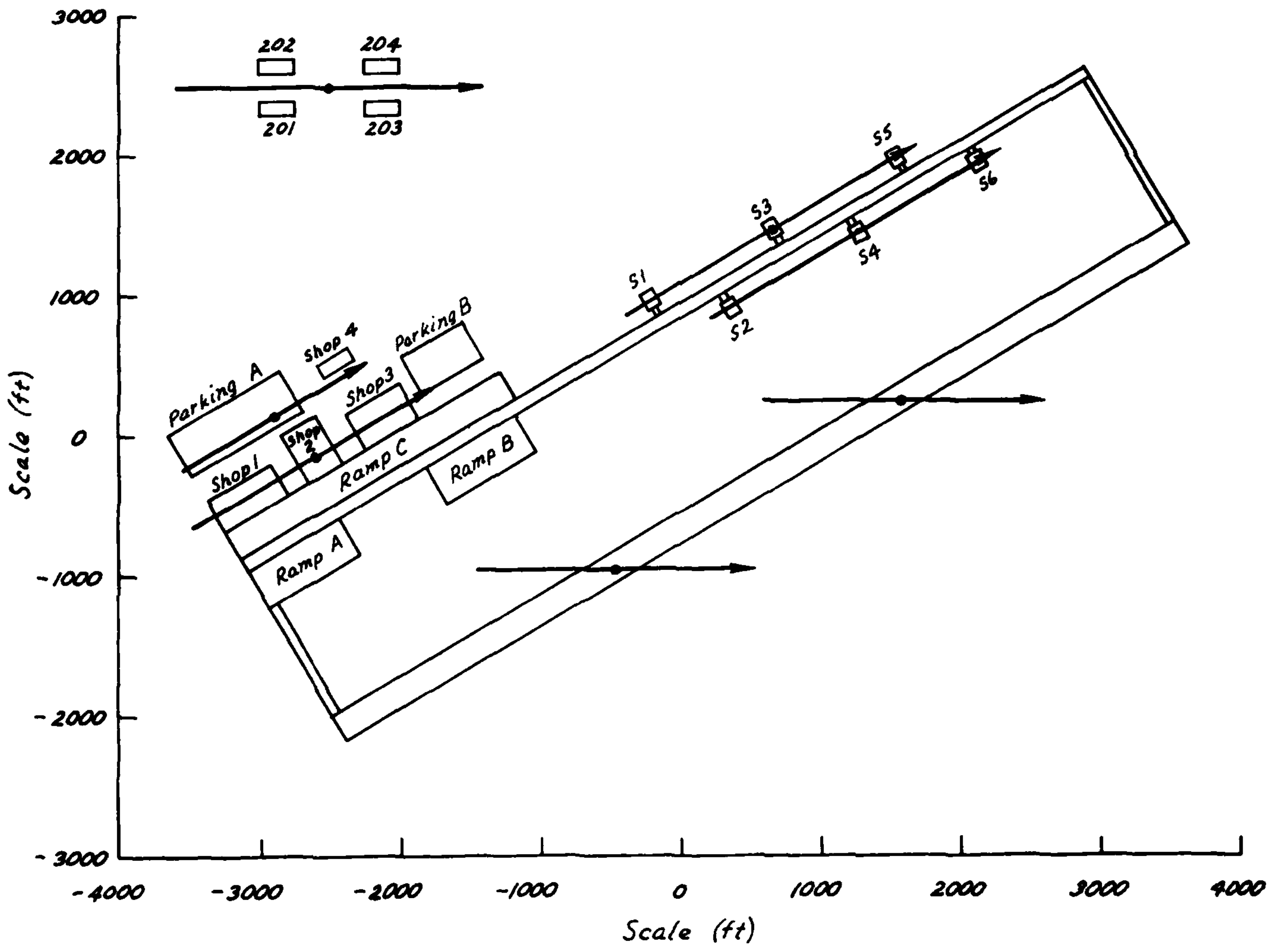


Fig. 1 — Test base

```
                Model Operations are Specified with the CONTrol card

CONT 1  0  8 10  1  1  3     0  4250    65     1     0    10   250       TESTBASE

                The DATA card controls the interactions with TSAR.

DATA        0     0     5     1     5    45     2     3     4    50

                The target data for these sample calculations are entered next.

TGT    -2500 -2000  7000   200    60     1                                RUNWAY
TGT    -3150  -880  7000    80    60     1                                MAINTXWY

TGT    -3380  -450   500   275    60     5           1              2     SHOP #1
      C  1     1   20 C  1     4    33 C  2     2    10
      C  1     2   18 C  1     3    35
TGT    -2860     0   275   450    60     6           2              1     SHOP #2
      C  1     2   30 C  2     3    25 C  2     4    50 C  1     1    35
TGT    -2390   150   425   300    60     5           3              1     SHOP #3
      C  1     1   18 C  1     2    22 C  1     3    40
TGT    -2600   500   250   100    60     6           4     1     7        SHOP #4

TGT    -3100  -960   750   300    60     4                                RAMP A
TGT    -1810  -240   750   300    60     4                                RAMP B
TGT    -3240  -700  2250   200    60     4                                RAMP C

TGT    -3650     0   900   300    60     7                          1     PKG AREA
      C  2     1   80 C  2     2    30 C  2     3    25 C  2     4    10
TGT    -2000   550   500   275    60     7                          1     PKG AREA
      C  2     1   20 C  2     2    50 C  2     3    50 C  2     4    40

TGT    -2950 -1240    50   850    60     3                                X TAXY 1
TGT     2840  2500    50   850    60     3                                X TAXY 2
TGT     -200   930    50    50    60     3                                STUB 1
TGT      310  1010    50    50    60     3                                STUB 2
TGT      670  1430    50    50    60     3                                STUB 3
TGT     1200  1530    50    50    60     3                                STUB 4
TGT     1550  1960    50    50    60     3                                STUB 5
TGT     2070  2050    50    50    60     3                                STUB 6

                The aircraft shelter data are entered with TGT type cards,
                rather then TGT2 type cards, since munitions and TRAP are
                stored in these facilties in this example.

TGT     -330  1070    75   125    60     2                          1     SHEL #1
      C  4     1  0.7 C  5     2   2.8
TGT      310   920    75   125    60     2                          1     SHEL #2
      C  4     3  3.1 C  5     4   4.3
TGT      530  1580    75   125    60     2                                SHEL #3
TGT     1200  1435    75   125    60     2                          1     SHEL #4
      C  4     3  3.1 C  5     4   4.3
TGT     1400  2090    75   125    60     2                          1     SHEL #5
      C  4     1  0.7 C  5     2   2.8 C  4     2   0.8 C  5     3   1.2
TGT     2090  1950    75   125    60     2                          1     SHEL #6
      C  4     2  2.8 C  5     3   1.2

                Various off-duty personnel are in the barracks.

TGT    -3000  2300   100   250     0     8                          1     BARRACKS #201
      C  1 1000    20 C  1  1003    44
TGT    -3000  2600   100   250     0     8                          1     BARRACKS #202
      C  1 1000    20 C  1  1003    22
TGT    -2250  2300   250   250     0     8                          1     BARRACKS #203
      C  1 1000    20 C  1  1003    34
TGT    -2250  2600   100   250     0     8                          1     BARRACKS #204
      C  1 1000    20

         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
```

Fig. 2 — TSARINA input-control data and target information

The attack data, equivalance data and the weapon effectiveness data are entered last.

```
ATT  3    90  -450  -950   300   150    75    55    24  2000     1    95
ATT  3    90  1600   250   300   150    75    55    24  2000     1    95
ATT  2    60 -2600  -150   300   150    75    55    24  2000     1    95
ATT  1    60 -2900   150   300   150    75    55    24  1500     1    95
ATT  4    60   650  1550   300   150    75    55    24  2500     1    95
ATT  4    60  1310  1400   300   150    75    55    24  2500     1    95
ATT  2    90 -2500  2500   300   150    75    55    24  2200     1    95
```

The equivalence data are used to assure compatability of resource types at the TSAR-TSARINA interface.

```
EQUI    5001     1    -1  5001     4     4     5  5004     1     1   101
EQUI       4   104  5005     2     2     1     7
EQUI    5001  1003  1003  1004  1005  5001     2   101     0
```

Fourteen cards are used to input the effectiveness data; direct hits and near hits are assessed with the same effectiveness factors, except for the eighth target type.

```
END 14  1 95    20     2    20    20    80   100    20   120
                     150          80   110   140    40    35
                      75                85    75    75 80 80
                      55                28    15    15 70 80
                      35      100 80    35    24    26 60 80
                      40                30    17    30
                      45                44    29    44
                                        11     0    11
                  33333         7   111545123120222545777
                     100                                 150
                                                          43
                      55                              100 80
                      69                               90 80
                      12                               80 80
END

         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
```

Fig. 3 — TSARINA input - attacks and weapons data

ordering of cards for a given case is generally immaterial except that (1) the CONTrol and DATA cards should be entered first, (2) all supplementary cards (used with the TGT, ATT2, and EMD cards) must be listed immediately following the card that they supplement, and (3) the EQUI cards should be entered as a group. The END card concludes the input stream and specifies the end of the assessment; alternatively, a REDO card can be entered to end the input for one assessment and call for a new case in which the attack is changed, or both the target complex and the attack are changed. In this sample we have not used either the TGT2 or ATT2 type cards; the TGT2 card types could have been used for the aircraft shelters if resources had not been stored in these facilities.

A careful review of these entries along with a reading of Appendix A should lead to a full and rapid understanding of the various input requirements. The CONTrol card data specify, for example, that ten trials are to be computed, and the runway availability is to be based on a 65 feet x 4250 feet minimum operating surface. Required repairs are to be assessed (the 1 in col. 45), but plots of the impact points are not to be generated (the 0 in col. 48). The DATA card specifies that TSAR is to simulate this attack at 5:45 am on the first day at base #1. Aircraft shelters, taxiways, and aircraft parking ramps are to be designated, respectively, as target types #2, #3, and #4.

As will be noted, the shop number has been entered in columns 53-54 of the TGT cards for the four shops, and the number of cards that are to be used to specify the resources that are associated with each target is specified in columns 68-70, except for shop #4. Since 100 percent of the on-duty type #7 personnel are in shop #4, columns 55-66 on the TGT card can be used to locate this resource. Shop #1 contains 20 percent of the type #1 personnel, 33 percent of the type #4 personnel, and 10 percent of the type #2 AGE. The larger vehicle parking area contains 80 percent of the type #1 AGE, and 30, 25, and 10 percent of AGE types #2, #3, and #4, respectively. The target data for the aircraft shelters and barracks are listed next; 0.7

percent of the type #1 munitions and 2.8 percent of the type #2 TRAP are in aircraft shelter #1; other shelters contain other quantities of munitions and TRAP. Eighty percent of the off-duty personnel are distributed in the four barracks, except for off-duty type #3 personnel who are located in the first three barracks.

The ATT cards in Figure 3 specify the attack headings, intended aim points, aiming errors, ballistic dispersion, number of weapons, bomb stick length, weapon type, and probability of arrival of the attacks. The EQUIvalence cards enter instructions for structuring the results to be transmitted to TSAR. The first entry specifies that type #1 personnel casualties are not to be reported to TSAR. The second entry prescribes that the percentage casualties to TSARINA type #4 personnel are to be imposed on types #4 and #5 personnel in TSAR. The next entry specifies that the percentage losses sustained by type #1 munitions are to be imposed on both assembled and unassembled #1 and #4 type munitions in TSAR; note that the "equivalence" list starts on the first EQUI card and is completed on the second. The last entry on the second card specifies that the casualty percentage sustained in TSARINA by the #2 type TRAP is to be imposed on three TSAR types of TRAP: #1, #2, and #7. The third EQUI card specifies that the casualties sustained by the off-duty type #3 personnel should be applied to the #3, #4, and #5 off-duty personnel types in TSAR. (Casualties among other off-duty types will be controlled by the losses to the type #1000 personnel who were located in the barracks.) The last entry on the third EQUI card specifies that the percentage loss sustained by personnel type #2 should be imposed on the aircrews (i.e., #101) at risk in TSAR.

The various weapon effectiveness data are entered with the EMD card and its supplementary cards. The first entry specifies that 14 cards are to be used to specify the weapon effects for this weapon. The second and third entries denote that the weapon is type #1 and that its reliability is 95 percent. The eight entries that follow are the radii of weapon effects against the eight target types found at the test base. The first of these specifies an effective disrupted radius of 20 feet on runways (target type #1).

The same value is indicated for the other pavement targets: e.g., the taxiways, aircraft parking ramps, and vehicle parking areas (target types #3, #4, and #7). Larger radii are specified for structural effects against the buildings (target types #5, #6, and #8).

The entries on the first supplementary EMD card specify the secondary weapon effects radii. The role and interpretation of these radii, and of the factors on the following six cards, are determined by the entries on the eighth supplementary card, as explained at length in Appendix A. For example, the values listed for the sixth type of target specify that personnel and munitions (or POL) that are within a radius of 100 feet, sustain 75 and 17 percent losses, respectively; i.e., their losses are governed by the first criterion. AGE and TRAP (the third and sixth supplementary cards) are governed by the second criterion, and 15 and 29 percent of these resources within 140 feet of a hit are lost. All spare parts within 140 feet of the burst sustain 24 percent losses, if the burst was within 100 feet of the target boundary (i.e., the third criterion). No assessment is provided for building materials in the sixth type of target.

The last three entries for the second target type--the aircraft shelters--are the three special factors that control aircraft and shelter damage estimates. The first two numbers are the probabilities that aircraft that are in the shelters at the time of the attack will be killed (1) if the shelter is buttoned up and a weapon strikes within the two foot radius entered on the first card, and (2) if the shelter door is open and a weapon strikes within the 100 foot radius noted on the tenth card. The last of these three entries specifies that there is a 12 percent chance that the shelter itself is destroyed, given a hit within two feet (R1) of the shelter walls.

The two-level cookie-cutter damage function is used for target type #8; when weapons fall outside the target, 80 percent of the personnel, equipment, and parts within the target are lost within 35 feet of an impact and 20 percent (one-forth of 80) of these resources are lost within 80, 70, and 60 feet from an impact, respectively.

If the weapon hits the target, 80 percent of the resources are lost within 43 feet of an impact, and 20 percent are lost within 100, 90, and 80 feet, respectively.

OUTPUT

The initial TSARINA output provides a record of the input data. The first of these data, shown in Figure 4, provides a record of all resources, including the number and name of the target at which the resources are stored. These are followed by the formatted title block shown in Figure 5 that indicates the values for several of the key control parameters. The scheduled time of the attack, and an indication of any coordinate translation that was required, are indicated next. Full particulars on the targets, attacks, and weapon effects data conclude the input data as shown in Figures 5 and 6. To distinguish aircraft shelters from other numbered facilities, a "1000" is added to the number for each shelter to avoid ambiguity.

TSARINA output for each trial is illustrated in Figures 7 and 8, using the results for the eighth trial. As will be noted, both the runway and the main taxiway were hit; of the 26 bombs that affected the runway, six did not impact the runway itself, but hit close enough for the runway to be within the bomb's 20 foot radius of effectiveness. Two of the six aircraft shelters each received one hit; in one case the impact was outside the target, but within the 2-foot radius of effectiveness. The fractional losses of any resources that were stored in each of these facilities are also noted.

The only hits that affect the taxiways, or the stub taxiways in front of the aircraft shelters, were on those stubs; stub #1 sustained a direct hit, and stub #5 sustained a near miss.

Three of the shops sustained several hits; the expected losses to the various classes of resources can be noted. For shop #2, for example, even though 5 out of the 9 hits were external to the facility, damage would be expected to nearly 70 percent of the structure, and 58 percent of the personnel in that facility are estimated to be lost.

```
                              RESOURCE STORAGE DATA

TARGET NUMBER          NAME

      3             SHOP #1              -3380    -450
                              C   1     1  200       C   1     4  330     C   2     2  100
                              C   1     2  180       C   1     3  350
      4             SHOP #2              -2860       0
                              C   1     2  300       C   2     3  250     C   2     4  500     C   1     1  350
      5             SHOP #3              -2390     150
                              C   1     1  180       C   1     2  220     C   1     3  400
      6             SHOP #4              -2600     500
                              C   1     7 1000
     10            PKG AREA              -3650       0
                              C   2     1  800       C   2     2  300     C   2     3  250     C   2     4  100
     11            PKG AREA              -2000     550
                              C   2     1  200       C   2     2  500     C   2     3  500     C   2     4  400
     20            SHEL #1                -330    1070
                              C   4     1    7       C   5     2   28
     21            SHEL #2                 310     920
                              C   4     3   31       C   5     4   43
     23            SHEL #4                1200    1435
                              C   4     3   31       C   5     4   43
     24            SHEL #5                1400    2090
                              C   4     1    7       C   5     2   28     C   4     2    9     C   5     3   12
     25            SHEL #6                2090    1950
                              C   4     2   28       C   5     3   12
     26            BARRACKS #201         -3000    2300
                              C   1  1000  200       C   1  1003  440
     27            BARRACKS #202         -3000    2600
                              C   1  1000  200       C   1  1003  220
     28            BARRACKS #203         -2250    2300
                              C   1  1000  200       C   1  1003  340
     29            BARRACKS #204         -2250    2600
                              C   1  1000  200

STORAGE OF THE RESOURCE LOCATIONS REQUIRED   55 LOCATIONS IN THE STOCKS ARRAY,
AND THE EQUIVALENCE DATA USED   21 ELEMENTS OF THE EQUIV ARRAY.
```

Fig. 4 — Listing of resource location data

*** TSARINA *** DAMAGE ASSESSMENTS FOR TSAR

BASED ON THE AIDA MODEL OF AIRBASE DAMAGE ASSESSMENT
DEVELOPED BY D. E. EMERSON AT THE RAND CORPORATION

NO OF TRIALS 10 NPRINT 0 DAMAGE 3 MODE 0 MCL 4250(250) MCW 65(10) MIN REPAIR 1 PLOT HITS 0 TEST 0

CASE # 1
IN THE TSAR SIMULATION, THIS ATTACK AND DAMAGE OCCUR AT BASE # 1 ON DAY 1 AT 5:45

***** BASE COMPLEX NAME - TESTBASE *****

ALL TARGET LOCATION DIMENSIONS WERE INCREASED BY 4000 IN THE X-DIMENSION AND 2000 IN THE Y-DIMENSION

TARGET DATA

NUMBER	X-DIM	Y-DIM	NE LIMB	SE LIMB	ANGLE	TGT TYPE	STORE	BLDG NO	
1	1500	0	7000	200	60	1	0	0	RUNWAY
2	850	1120	7000	80	60	1	0	0	MAINTXWY
		** TARGET TYPE #	2 **						
20	3670	3070	75	125	60	2	0	1001	SHEL #1
21	4310	2920	75	125	60	2	0	1002	SHEL #2
22	4530	3580	75	125	60	2	0	1003	SHEL #3
23	5200	3435	75	125	60	2	0	1004	SHEL #4
24	5400	4090	75	125	60	2	0	1005	SHEL #5
25	6090	3950	75	125	60	2	0	1006	SHEL #6
		** TARGET TYPE #	3 **						
12	1050	760	50	850	60	3	0	0	X TAXY 1
13	6340	4500	50	850	60	3	0	0	X TAXY 2
14	3800	2910	50	50	60	3	0	0	STUB 1
15	4310	3010	50	50	60	3	0	0	STUB 2
16	4670	3430	50	50	60	3	0	0	STUB 3
17	5200	3530	50	50	60	3	0	0	STUB 4
18	5550	3960	50	50	60	3	0	0	STUB 5
19	6070	4050	50	50	60	3	0	0	STUB 6
		** TARGET TYPE #	4 **						
7	900	1040	750	300	60	4	0	0	RAMP A
8	2190	1760	750	300	60	4	0	0	RAMP B
9	760	1300	2250	200	60	4	0	0	RAMP C
		** TARGET TYPE #	5 **						
3	620	1550	500	275	60	5	0	1	SHOP #1
5	1610	2150	425	300	60	5	0	3	SHOP #3
		** TARGET TYPE #	6 **						
4	1140	2030	275	450	60	6	0	2	SHOP #2
6	1400	2500	250	100	60	6	0	4	SHOP #4
		** TARGET TYPE #	7 **						
10	350	2300	900	300	60	7	0	0	PKG AREA
11	2000	2550	500	275	60	7	0	0	PKG AREA
		** TARGET TYPE #	8 **						
26	1000	4300	100	250	0	8	0	0	BARRACKS #201
27	1000	4600	100	250	0	8	0	0	BARRACKS #202
28	1750	4300	250	250	0	8	0	0	BARRACKS #203
29	1750	4600	100	250	0	8	0	0	BARRACKS #204

Fig. 5 — Listing of control and target data

ATTACK DATA

NUMBER	HDG	X-DMPI	Y-DMPI	REP	DEP	R-DISP	D-DISP	NO WPNS	LENGTH	WPN TYPE	ARRIVAL
1	90	3550	1050	300	150	75	55	24	2000	1	95
2	90	3550	1050	300	150	75	55	24	2000	1	95
3	90	3550	1050	300	150	75	55	24	2000	1	95
4	90	5600	2250	300	150	75	55	24	2000	1	95
5	90	5600	2250	300	150	75	55	24	2000	1	95
6	90	5600	2250	300	150	75	55	24	2000	1	95
7	60	1400	1850	300	150	75	55	18	2000	1	95
8	60	1400	1850	300	150	75	55	18	2000	1	95
9	60	1100	2150	300	150	75	55	18	1500	1	95
10	60	4650	3550	300	150	75	55	18	2500	1	95
11	60	4650	3550	300	150	75	55	18	2500	1	95
12	60	4650	3550	300	150	75	55	18	2500	1	95
13	60	4650	3550	300	150	75	55	18	2500	1	95
14	60	5310	3400	300	150	75	55	18	2500	1	95
15	60	5310	3400	300	150	75	55	18	2500	1	95
16	60	5310	3400	300	150	75	55	18	2500	1	95
17	60	5310	3400	300	150	75	55	14	2500	1	95
18	90	1500	4500	300	150	75	55	18	2200	1	95
19	90	1500	4500	300	150	75	55	18	2200	1	95

MISS DISTANCE AND KILL PROBABILITY DATA

WPN TYPE	WPN REL	TARGET TYPES 1	2	3	4	5	6	7	8	9	10
1	0.950	20	2	20	20	80	100	20	120	0	0
		0	150	0	80	110	140	40	35	0	0
		0	75	0	0	85	75	75	80080	0	0
		0	55	0	0	28	15	15	70030	0	0
		0	35	0	100080	35	24	26	60040	0	0
		0	40	0	0	30	17	30	0	0	0
		0	45	0	0	44	29	44	0	0	0
		0	0	0	0	11	0	11	0	0	0
		0	333330	0	7000	111545	123120	222545	777000	0	0
		-1	100	-1	-1	-1	-1	-1	150	-1	-1
		0	0	0	0	0	0	0	42	0	0
		0	55	0	0	0	0	0	100080	0	0
		0	68	0	0	0	0	0	90090	0	0
		0	12	0	0	0	0	0	90090	0	0

Fig. 6 — Listing of attack and weapons data

***** TRIAL # 8 *****

TARGET HIT SUMMARY TRIAL 8

TGT NO	NUMBER HITS TOT	NUMBER HITS OUT	CPU COVERAGE	COVERAGE R1	COVERAGE R2	PEOPLE	AGE	PARTS	AMMO	TRAP	MATERL	BLDG NO	
			** TARGET TYPE # 1 **										
1	26	6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	RUNWAY
2	2	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	MAINTXWY
			** TARGET TYPE # 2 **										
20	1	0	0.0	0.001	1.000	0.750	0.550	0.350	0.400	0.450	0.0	1001	SHEL #1
21	0	0	0.0	0.0	1.000	0.0	0.0	0.0	0.0	0.0	0.0	1002	SHEL #2
22	0	0	0.0	0.0	0.956	0.0	0.0	0.0	0.0	0.0	0.0	1003	SHEL #3
23	1	1	0.0	0.000	1.000	0.750	0.550	0.350	0.400	0.450	0.0	1004	SHEL #4
24	0	0	0.0	0.0	0.533	0.0	0.0	0.0	0.0	0.0	0.0	1005	SHEL #5
25	0	0	0.0	0.0	0.818	0.0	0.0	0.0	0.0	0.0	0.0	1006	SHEL #6
			** TARGET TYPE # 3 **										
14	1	0	0.0	0.443	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	STUB 1
18	1	1	0.0	0.389	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	STUB 5
			** TARGET TYPE # 4 **										
9	9	1	0.0	0.022	0.281	0.0	0.0	0.256	0.0	0.0	0.0	0	RAMP C
			** TARGET TYPE # 5 **										
3	6	3	0.0	0.520	0.754	0.461	0.179	0.219	0.099	0.197	0.014	1	SHOP #1
5	0	0	0.0	0.0	0.006	0.0	0.0	0.0	0.0	0.0	0.0	3	SHOP #3
			** TARGET TYPE # 6 **										
4	9	5	0.0	0.696	0.912	0.580	0.266	0.394	0.169	0.457	0.0	2	SHOP #2
6	4	2	0.0	0.906	0.999	0.784	0.334	0.487	0.241	0.567	0.0	4	SHOP #4
			** TARGET TYPE # 7 **										
10	8	1	0.0	0.033	0.127	0.097	0.020	0.035	0.073	0.153	0.010	0	PKG AREA
			** TARGET TYPE # 8 **										
26	5	4	0.0	0.835	0.143	0.190	0.171	0.156	0.0	0.0	0.0	0	BARRACKS #201
27	3	2	0.0	0.918	0.169	0.246	0.226	0.204	0.0	0.0	0.0	0	BARRACKS #202
28	3	2	0.0	0.482	0.053	0.081	0.073	0.066	0.0	0.0	0.0	0	BARRACKS #203
29	4	4	0.0	0.838	0.069	0.143	0.115	0.091	0.0	0.0	0.0	0	BARRACKS #204

Fig. 7 — TSARINA trial results

TSARINA RESOURCE LOSS FRACTIONS FOR TRIAL #8

CLASS 1	TYPE	1	0.295
CLASS 1	TYPE	2	0.257
CLASS 1	TYPE	3	0.161
CLASS 1	TYPE	4	0.152
CLASS 1	TYPE	7	0.784
CLASS 1	TYPE	1003	0.165
CLASS 1	TYPE	1000	0.132
CLASS 2	TYPE	1	0.016
CLASS 2	TYPE	2	0.024
CLASS 2	TYPE	3	0.071
CLASS 2	TYPE	4	0.135
CLASS 4	TYPE	1	0.003
CLASS 4	TYPE	2	0.0
CLASS 4	TYPE	3	0.012
CLASS 5	TYPE	2	0.013
CLASS 5	TYPE	3	0.0
CLASS 5	TYPE	4	0.019

LOSSES TO AIRCRAFT, SUPPORT RESOURCES AND FACILITIES -- FORMATTED FOR TRANSFER TO THE TSAR SIMULATION MODEL

40	5	1	5	45	0	0	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	1	101	26	0	0	0	1	3	16	0	0	0
40	0	0	0	0	1	4	15	0	0	0	1	5	15	0	0	0
40	0	0	0	0	1	7	78	0	0	0	1	1003	17	0	0	0
40	0	0	0	0	1	1004	17	0	0	0	1	1005	17	0	0	0
40	0	0	0	0	1	1000	17	0	0	0	0	0	0			
40	0	0	0	0	2	1	2	0	0	0	2	2	2	0	0	0
40	0	0	0	0	2	3	7	0	0	0	2	4	13	0	0	0
40	0	0	0	0	4	1	0	0	0	0	4	101	0	0	0	0
40	0	0	0	0	4	4	0	0	0	0	4	104	0	0	0	0
40	0	0	0	0	4	3	1	0	0	0						
40	0	0	0	0	5	2	1	0	0	0	5	1	1	0	0	0
40	0	0	0	0	5	7	1	0	0	0	5	4	2	0	0	0
40	0	0	0	0	9	1	52	44	18	22	9	2	70	58	27	39
40	0	0	0	0	9	4	91	78	33	43	0	0	0	0	0	0
40	0	0	0	0	9	60	1	0	0	0	0	0	0	0	0	0
40	0	0	0	0	9	59	2	0	0	0						
40	0	0	0	0	10	1	4	91	12	0	10	2	50			
40	0	0	0	0	8	0	18	100	100	64						
0																

Fig. 8 — Trial results of fractional losses

The fractional losses for all of the resource types present on the base are summarized in Figure 8; these results cumulate the fractional losses suffered by each type of resource at each of its locations. These same results, modified as required by the EQUI cards, and summary data describing the damage sustained by the runway, taxiways, aircraft shelters, and other facilities, are shown at the bottom of Figure 8, formatted for transfer to TSAR. To interpret these card images, the reader should consult the instructions for preparation of Input Card Type #40 in the TSAR User's Manual.

The statistical results for the ten trials are presented in Figures 9 and 10. The first of these figures provides an indication of the fraction of trials in which at least one hit was sustained, as well as the expected number, and standard deviation, of hits. Comparable results are provided for CBU weapons, when they have been used. The results labelled "Bomb Coverage" are the expected fractions of the facility floor-space that are affected by the two coverage radii, R1 and R2. The average losses that are sustained at each target by the six classes of resources are listed on the right side of Figure 9. A summary of the runway closures and the required runway repairs is noted at the bottom of this figure.

Figure 10 presents the average losses sustained by each type of resource for the ten trials, along with the standard deviation of those losses. When TSARINA is used as a general-purpose damage assessment model, these statistical results are the primary output; they are not transferred to TSAR, since TSAR only uses the trial-by-trial results illustrated in Figure 9.

TARGET DAMAGE STATISTICS FOR 10 TRIALS

TARGET NO	PERCENT ATTACKS HIT	AVERAGE HITS PER ATTACK	STD DEV OF HITS	AVG CBU COVERAGE	STD DEV COVERAGE	BOMB COVERAGE P1	P2	KILL PROBABILITIES PEOPLE	AGE	PARTS	AMMO	TRAP	MATERL	BLDG NO/NAME
TARGET TYPE # 1														
1	100.0	29.80	6.68	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	RUNWAY
2	50.0	21.40	12.55	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	MAINTXWY
TARGET TYPE # 2														
20	50.0	0.50	0.53	0.0	0.0	0.001	0.983	0.375	0.275	0.175	0.200	0.225	0.0	SHEL #1
21	40.0	0.40	0.52	0.0	0.0	0.000	0.890	0.300	0.220	0.140	0.160	0.180	0.0	SHEL #2
22	30.0	0.30	0.48	0.0	0.0	0.000	0.986	0.225	0.165	0.105	0.120	0.135	0.0	SHEL #3
23	40.0	0.40	0.52	0.0	0.0	0.000	0.930	0.300	0.220	0.140	0.160	0.180	0.0	SHEL #4
24	20.0	0.20	0.42	0.0	0.0	0.000	0.952	0.150	0.110	0.070	0.080	0.090	0.0	SHEL #5
25	40.0	0.40	0.52	0.0	0.0	0.001	0.892	0.300	0.220	0.140	0.160	0.180	0.0	SHEL #6
	36.67	0.37												
TARGET TYPE # 3														
13	20.0	0.20	0.42	0.0	0.0	0.005	0.0	0.0	0.0	0.0	0.0	0.0	0.0	X TAXY 2
14	20.0	0.40	0.97	0.0	0.0	0.080	0.0	0.0	0.0	0.0	0.0	0.0	0.0	STUB 1
15	10.0	0.10	0.32	0.0	0.0	0.001	0.0	0.0	0.0	0.0	0.0	0.0	0.0	STUB 2
16	30.0	0.30	0.48	0.0	0.0	0.056	0.0	0.0	0.0	0.0	0.0	0.0	0.0	STUB 3
17	20.0	0.20	0.42	0.0	0.0	0.016	0.0	0.0	0.0	0.0	0.0	0.0	0.0	STUB 4
18	10.0	0.10	0.32	0.0	0.0	0.009	0.0	0.0	0.0	0.0	0.0	0.0	0.0	STUB 5
19	40.0	0.50	0.71	0.0	0.0	0.072	0.0	0.0	0.0	0.0	0.0	0.0	0.0	STUB 6
TARGET TYPE # 4														
8	0.0	0.0	0.0	0.0	0.0	0.0	0.005	0.0	0.0	0.005	0.0	0.0	0.0	RAMP B
9	70.0	6.30	7.75	0.0	0.0	0.014	0.182	0.0	0.0	0.171	0.0	0.0	0.0	RAMP C
TARGET TYPE # 5														
3	100.0	7.10	2.19	0.0	0.0	0.422	0.612	0.377	0.150	0.183	0.085	0.169	0.012	1 SHOP #1
5	90.0	7.20	5.47	0.0	0.0	0.409	0.576	0.368	0.150	0.186	0.093	0.166	0.012	3 SHOP #3
TARGET TYPE # 6														
4	100.0	4.70	2.54	0.0	0.0	0.637	0.843	0.535	0.254	0.372	0.161	0.432	0.0	2 SHOP #2
6	60.0	1.50	1.65	0.0	0.0	0.321	0.497	0.264	0.126	0.178	0.074	0.223	0.0	4 SHOP #4
TARGET TYPE # 7														
10	70.0	4.70	4.27	0.0	0.0	0.020	0.076	0.058	0.012	0.021	0.044	0.090	0.006	PKG AREA
11	70.0	1.80	1.62	0.0	0.0	0.015	0.055	0.042	0.009	0.015	0.032	0.066	0.005	PKG AREA
TARGET TYPE # [illegible]														
26	90.0	2.40	2.30	0.0	0.0	0.061	0.126	0.171	0.158	0.164	0.0	0.0	0.0	BARRACKS #231
27	100.0	3.50	1.38	0.0	0.0	[illegible]	0.074	0.145	0.128	0.113	0.0	0.0	0.0	BARRACKS #232
28	90.0	4.50	2.30	0.0	0.0	[illegible]	0.085	0.134	0.118	0.104	0.0	0.0	0.0	BARRACKS #233
29	100.0	3.70	1.70	0.0	0.0	[illegible]	0.144	0.215	0.193	0.172	0.0	0.0	0.0	BARRACKS #234

AT LEAST ONE MINIMUM RUNWAY SECTION WAS OPEN AFTER 10.0 PERCENT OF THE ATTACKS
WHEN ALL RUNWAYS WERE CLOSED, 2.67(1.80) HOURS [illegible] REPAIR, [illegible] THE AVERAGE, TO PROVIDE A MINIMUM RUNWAY

Fig. 9 — Statistical results of target damage

RESOURCE CLASS	TYPE	AVERAGE LOSSES	STD DEV LOSSES
		PERCENT	
PEOPLE	1	32.830	14.394
	2	30.890	14.016
	3	27.840	16.968
	4	12.390	7.466
	7	26.330	31.526
	1000	13.260	6.380
	1003	15.230	11.599
AGE	1	1.080	0.932
	2	2.240	1.154
	3	7.030	2.788
	4	13.130	5.206
AMMO	1	0.150	0.220
	2	0.500	0.548
	3	0.960	0.824
TRAP	2	0.850	0.883
	3	0.300	0.376
	4	1.520	1.261

Fig. 10 — Statistical results of resource losses

Appendix A

DETAILED DESCRIPTION OF TSARINA INPUT

The basic input cards employed with TSARINA are:

CONT	control card
DATA	TSAR data card
TGT	target card; one per target
TGT2	aircraft shelter card; one per shelter when the expected-value mode is to be used
ATT	attack card; one per weapon delivery pass (or group of identical passes)
ATT2	alternate attack card
EMD	effective miss distance card; one for each weapon type
REDO	controls sequential cases
END	terminates overall computation

The ATT2 card is actually two cards in sequence and the EMD card may have up to sixteen* supplementary cards. Each TGT card is followed by as many cards as are required to specify the resources located at that target. A detailed description of the entries for each type of card is presented on the pages that follow.

The general arrangement of data on all basic card types is similar; the card name is placed (left-adjusted) in the first four columns and the data are entered in the eleven 6-column fields between columns 7 and 72. All data are read with an I6 format, i.e., they are integers, except that, as will be noted

*Up to 33 supplementary cards, if there are more than ten target types.

from the descriptions defining data entry, two data are entered in certain fields of the CONT and DATA cards and on the supplementary target cards. Columns 5 and 6 are also used on several cards, as will be described. Furthermore, the name of the target complex being studied and a name for each target may be included in columns 73 through 80 of the CONT and TGT* cards, respectively; any alphanumeric names are acceptable.

All linear dimensions should be in consistent units** (e.g., feet) and the target orientation and the attack heading entries should be in degrees.

---

*When electronic card images are used, columns 73 through 88 may be used for alphanumeric target names by entering a "1" in column 18 of the CONT card.

**If ATT2 cards are to be used, all linear dimensions must be in feet.

CONT

The CONT card controls the mode of operation, the choice of random number generator, the number of trials (attack replications), and printout options; it also specifies the minimum clear length (MCL) and minimum clear width (MCW) for runway attack effectiveness calculations, and controls runway repair assessment. This card should be the first card to be entered.

Columns	Data Entry
1-4	CONT
6	If unity, program computes resource damage levels appropriate for entry into TSAR.
8-9	When 0, the seed for the random number generator is the same for all runs. If greater than 0, the seed is changed from run to run; if equal to -1, the random number generator is locked out.
10-12	Number of target types to be entered.
13-15	Desired number of replications. Default is 1.
16-18	If 1, descriptive data on the CONT and TGT cards may extend to column 88, rather than be constrained to an 80-column format.
19-21	Controls printout options as follows: If entry is: 5 Prints multiple trial statistics plus a condensed listing of hits by trial 4 Prints multiple trial statistics plus a condensed listing of runway status by trial 3 Prints multiple trial statistics only 2 All above plus runway results for each trial 1 All above plus hit summary for each trial -1 All above plus all hits and target corners -2 All above plus all impact points
23-24	Controls printout options for resource damage: 1 Damage fraction formatted only for user 2 Damage fraction formatted only for TSAR 3 Both formats

29-30 Normally set to zero; when initialized greater than zero, intermediate computational information is output for program test purposes. If set to greater than 7, the random number generator is locked out. See the program source listing in Appendix C for the effect of other values.

31-36 Minimum clear length (MCL) for aircraft operations. (Used to test if the runways are open.)

37-42 Minimum clear width (MCW) for aircraft operations. (Used to test if the runways are open.)

45 When the entry is 1, runway results will include the minimum number of craters to be repaired for the runway to meet the MCL and MCW criteria.

48 When the entry is 1, a plot of all [illegible] will be included for all [illegible] printout option entry [illegible] than 33; when the entry is 2, [illegible] vided for each runway [illegible]

49-54 The distance across the runway that the "[illegible] runway rectangle" is to be shifted in [illegible] an adequate section; the default value is [illegible]

55-60 The distance along the runway that the minimum runway rectangle is to be shifted in checking for an adequate section; the default value is 250.

73-80 A name can be entered here for the entire target complex and it will appear in the heading of the output listing.

DATA

The DATA card controls the form of the output to TSAR, defines the time and location of the attack, and provides TSARINA the necessary resource identity data for communicating results to TSAR. This card is not required if the results are not to be used in TSAR.

Columns	Data Entry
1- 4	DATA
5- 6	If unity, statistical results are punched on cards for subsequent processing and reorganization using auxiliary programs.
12	Enter the number of trials for which damage data and resource loss data are to be stored for TSAR.
18	Enter the number of trials for which card copies of the damage data and resource loss data are to be punched.
24	Number of airbase under attack (as interpreted within TSAR).
29-30	Day of attack
35-36	Hour of attack
41-42	Minute of attack
47-48	Target type number assigned to aircraft shelters
53-54	Target type number assigned to aircraft taxiways
59-60	Target type number assigned to aircraft parking ramps
61-66	Percentage of the aircraft that are damaged by air attack that cannot be repaired.

TGT

Each TGT card designates the location, size, and orientation of a rectangular target.

Columns	Data Entry
1-3	TGT
7-12	The X-coordinate of the westernmost corner of the target. If the westernmost corner of any target does not fall in the first quadrant of the X-Y coordinate system, TSARINA automatically translates the origin (a multiple of 1000 feet) so that all targets are in the first quadrant. If after translation, targets do not fall within the allowed 32000 x 32000 area, they are "moved" to the edge of that area and the user is notified.
13-18	The Y-coordinate of the westernmost corner of the target. If a target boundary runs exactly north-south, the X and Y coordinates of the southwestern corner should be specified.
19-24	Target dimension along the boundary running northeast (or north) from the reference corner specified in the two previous fields.
25-30	Target dimension along the boundary running southeast (or east) from the reference corner.
31-36	Heading in degrees of the northeast (or north) heading boundary of the target (along the dimension specified in columns 19 to 24).
41-42	Target type. Targets may be grouped into up to 10 (or 20) different categories with like vulnerabilities. This entry is used in conjunction with the effective miss distance on the EMD card. Target type #1 is restricted to runways and taxiways that may be used for flight operations; there can be at most 5 targets of this type. The user may specify other target types as aircraft parking ramps, taxiways, and aircraft shelters; if used with TSAR, the target type number selected to designate each of these target sets must be entered on the DATA card. For all other targets, structures with materiel of like vulnerability can be assigned a common target type number; if additional stratification in results is desired, targets of like vulnerability may be grouped under two or more target types.

48	If greater than zero, all hit locations will be saved and printed when the entry in column 24 of the CONT card is zero or less.
49-54	Facility number as understood in TSAR, if the target is a maintenance shop or a flight-line personnel assembly area (should not be used if the results are not to be used in TSAR).
55-60	Class of resource stored in the facility, if storage is restricted to 100 percent of one resource class, or to 100 percent of one type of one resource class.
61-66	Type of resource, if only one type is stored in the facility (all types are inferred if blank or zero).
67-70	Number of subsequent cards used to describe the types and quantities of resources stored in this facility (use only when columns 55-66 are blank).
73-80	Target descriptions; columns 73-88 may be used if input is not restricted to 80 columns.

SUPPLEMENTARY TARGET CARDS

Each TGT card may be followed by as many supplementary cards as are necessary to define the resources that are located in that particular target. Each of these cards is read with a 5X, 5(3X,I2, I5, F5.0) format that provides for five descriptions of resource class, resource type, and percent at the target. Entry of the letter "C" (for class), preceding the resource class (in columns 7, 22, 37, 52, and 67), has been found helpful in reviewing the large data sets required to represent a complex airbase (see Figure 2).

Columns	Data Entry
9-10 24-25 39-40 54-55 69-70	Number identifying the resource class: 1 Personnel — 5 TRAP 2 AGE and equipment — 6 Building materials 3 Aircraft parts — 7 POL 4 Munitions
11-15 26-30 41-45 56-60 71-75	Number identifying which type of the specified resource class is located here.* If there is no entry for "type", all types of the specified class (that have not otherwise been specified) are assumed to be present.
16-20 31-35 46-50 61-65 76-80	Percentage of the base stocks of the specified type and class of resource that are located in this target. Whole numbers are interpreted as percentages; a decimal entry is required to specify tenths of a percent. The output listing of resource storage data is in tenths of percent; e.g., 273 implies 27.3 percent.

---

*The number "1000" added to a personnel designation specifies off-duty personnel, and "100" added to a munitions designator specifies weapons that are not assembled.

TGT2

These cards are used to input the location of aircraft shelters when resources are not tc be located in the shelters and the damage to these targets is to be estimated with the expected-value mode. The identification "TGT2" designates selection of this option. All aircraft shelters must be handled in a consistent fashion; i.e. their characteristics must either all be entered on TGT type cards, or all entered on TGT2 type cards. The inputs for the TGT2 cards are identical with the TGT cards for columns 7-48; entries in columns 49-80 are ignored.

ATT

The ATT card specifies the parameters of each weapon-delivery pass. Inputs required are the attack heading (measured from north in the coordinate system used to specify the targets), the desired mean point of impact (DMPI) for a single weapon or for the middle of a stick of weapons, the aiming error expressed as REP (range error probable) and DEP (deflection error probable), the ballistic error of the individual weapons, the number of weapons to be delivered in the pass, the stick length, the weapon type (related to the effective miss distance on the EMD card), and the probability of arrival at the target.

Columns	Data Entry
1-3	ATT
5-6	Total number of passes with identical characteristics; default = 1.
10-12	Attack heading; degrees from north.
13-18	The X-coordinate of the DMPI of a single weapon or the middle of a stick of weapons.
19-24	The Y-coordinate of the DMPI as above.
25-30	The REP
31-36	The DEP
37-42	Ballistic dispersion in range of individual weapons (R-DISP).
43-48	Ballistic dispersion in deflection of individual weapons (D-DISP). Default value is R-DISP.
49-54	The number of weapons in the stick.
55-60	The length of the stick (the distance between the first and last weapon of the stick in the absence of dispersion).
61-66	The weapon type (provides reference to the appropriate effectiveness data). An entry is required (an integer from 1 to 10); otherwise hits will not be recorded.
67-72	Probability of arrival at target; default = 100.

ATT2

The ATT2 card should be used in place of the ATT card when the user wishes assistance with trajectory calculations. With this card the user expresses the attack in terms of speed, altitude, dive angle, intervalometer settings, etc., and a special subroutine converts these inputs to those demanded on the ATT card. The conversion procedure is the JMEM/AS Open-End Method Zero as outlined in the Users' Manual for JMEM/AS Open-End Methods, Wang Labs., Inc., Tewksbury, Mass., August 1974.

Both ATT and ATT2 type cards may be used in the same run; the order of entry is of no importance. When ATT2 cards are used, the input data will be reproduced as submitted, as well as being tabulated in the normal manner, after conversion.

Data input with the ATT2 procedure require two cards. The first card is labeled ATT2 in the first 4 columns and has input similar to that on an ATT card (all fields are read with an I6 format); a second unlabeled card is mandatory following each ATT2 card. The format for both cards follows. When these cards are used, all linear dimensions in the input data must be in feet.

Columns	Data Entry
1-4	ATT2
5-6	Total number of passes with identical characteristics; default = 1.
10-12	Attack heading in degrees from north.
13-18	The X-coordinate of the desired mean point of impact (DMPI) of a single weapon or the middle of a stick of weapons.
19-24	The Y-coordinate of the DMPI as above.
25-30	The CEP in the normal plane in mils, or, if DEP is specified, a constant which, when divided by the sine of the impact angle, gives the REP, in mils.
31-36	The DEP in mils (if omitted, CEP controls).

37-42	Ballistic dispersion in mils.
49-54	The number of weapons in the stick.
61-66	The weapon type.
67-72	Probability of arrival at target; default = 100.

The data format for the second card of each ATT2 pair is as noted below (this card is used with a 6X, 9I6 format). Typical ballistic data required for this card are noted in Table A-1.

Columns	Data Entry
7-12	Aircraft velocity (kn).
13-18	Release altitude of last bomb (ft).
19-24	Dive angle at release (deg).
25-30	Terminal velocity of a low-drag weapon, or the first leg of a high-drag bomb (ft/sec) (See Table A-1) (VT1 in JMEM).*
31-36	Terminal velocity of a cluster bomblet or a high-drag bomb (ft/sec) (See Table A-1) (VT2 in JMEM).
37-42	Probable error in estimating and correcting for wind effects (ft/sec).
43-48	Cluster opening time or fin opening time for a high-drag bomb (msec) (TD in JMEM).
49-54	Intervalometer setting (msec).
55-60	Dispensor intervalometer setting (msec) (0 for clusters).

*Illustrative values are noted on Table A-1.

Table A-1

TYPICAL BALLISTIC PARAMETERS

Weapon	VT1 (fps)	VT2 (fps)	T or H
Mk-81 Mod 1	1850	0	0
Mk-81 SE	1100	208	300 msec
Mk-82 Mod 1	1900	0	0
Mk-82 SE	1200	240	350 msec
Mk-83	2250	0	0
Mk-84	2850	0	0
M-117 Unretarded	1950	0	0
M-117 Retarded	900	168	300 msec
M-118	2450	0	0
AN-M64A1	1600	0	0
AN-M65A1	2000	0	0
Mk-36 DST	1200	0	350 msec
CBU-38	450	0	0
CBU-52B/B	1000	230	Variable altitude (ft)
CBU-58/B	950	215	Variable altitude (ft)

SOURCE: Users' Manual for JMEM/AS Open-End Methods, Wang Labs, Inc., Tewksbury, Mass., August 1974.

EMD

The EMD and supplementary cards provide information regarding weapon effectiveness against the several types of targets and the several classes of resources. The formats of the entries differ for point-impact weapons and for CBU munitions, and they differ between aircraft shelters and all other target types. Normally, each type of weapon will be represented by up to 17 cards (or 34 cards, if 11-20 target types are specified on the CONTrol card), although just the first card, the EMD card, could suffice for certain limited assessments.

For point-impact weapons (GP bombs or PGMs) the entries on the EMD card are:

Columns	Data Entry
1-3	EMD
5-6	Enter the maximum number of cards that are associated with each target type for this type of weapon.
8-9	Weapon type number
10-12	Weapon reliability (percentage)
13-18	R1 Radius of effectiveness versus target type #1
19-24	R1 Radius of effectiveness versus target type #2
o	
o	
o	
67-72	R1 Radius of effectiveness versus target type #10

The supplementary data for the several target types are located in the corresponding fields on the cards that immediately follow the first card. Definitions of the data to be entered on the EMD card and on the supplementary cards are noted below; somewhat different definitions apply for the target type that the user has designated as aircraft shelters, as will be outlined shortly.

Card No.		All Target Types (except shelters)
1	10	R1 Effective radius against target type
2	11	R2 Secondary effects radius
3	12	Personnel loss criteria
4	13	AGE loss criteria
5	14	Spare parts loss criteria
6	15	Munitions (and POL) loss criteria
7	16	TRAP loss criteria
8	17	Building materials loss criteria
9		Flag - Controls loss criteria interpretation

The data on cards 1 to 8 apply in the case of a near miss and those on cards 10 to 17 apply for a direct hit. If no data are entered for a direct hit, the near miss inputs are used; if any data are entered for a direct hit, only the values on the tenth through seventeenth cards are used. Null entries are interpreted as zero.

If 11 to 20 target types are treated, a second set of (up to 17) cards should be placed immediately after the first set described above; these cards are each read with a 12X, 10I6 format.

The appropriate loss criterion for assessing resource damage is controlled by the user and is communicated with the value of the control variable Flag, which is entered on the ninth card. The value of Flag for each target type is a 6-digit code UVWXYZ, where U, V, W, X, Y, and Z express the user's choices regarding the treatment of the six resource classes. Their values are defined from 0 to 8 as:

0	Ignore this class of resource
1	Value represents probability of kill of these resources within a circle of radius R1
2	Value represents probability of kill of these resources within a circle of radius R2
3	As in 2, given that the R1 radius intersects the target perimeter.
4	Value is the radius of kill of these resources
5	Value is the radius of kill of these resources, given that the R1 radius intersects the target perimeter

Values 6, 7, and 8 are a combination of a radius and a probability of kill.

6	Value is the radius (times 1000) of an area within which the Pk is one-fourth that value of Pk within R1, plus the value of Pk within R1
7	Value is the radius (times 1000) of an area within which the Pk is one-fourth that value of Pk within R2, plus the value of Pk within R2
8	As in 7, given that the radius R1 intersects the target perimeter.

Thus, Flag = 321475 implies that personnel, AGE, parts, munitions (or POL), TRAP, and building materials are to be assessed by options 3, 2, 1, 4, 7, and 5, respectively, for the particular weapon type and target type for which it is listed.

The use of the EMD cards is somewhat different for aircraft shelters. When the Monte Carlo mode is used to assess damage to these targets, the second through ninth EMD cards are interpreted the same as for any other target type. When the expected value mode is used, only the third, fourth, and fifth of these cards are used, and the entries are interpreted as the damage probability to personnel, equipment, and spare parts when a weapon falls within a radius R1 from the shelter. The other entries, noted below, are interpreted identically in either mode.

Card No.	Entry
1	R1 Effective radius against aircraft in shelters with closed doors.
10	R3 Effective radius against aircraft in shelters with open doors.
12	Probability of aircraft damage in a shelter with a closed door, when a weapon falls within a radius of R1 from the shelter.
13	Probability of aircraft damage in a shelter with an open door, when a weapon falls within a radius of R3 from the shelter.
14	Probability the aircraft shelter is killed when a weapon strikes within a radius of R1 from the shelter.

If the weapons are CBU type munitions, the following entries are used with the EMD and the supplementary cards:

Columns	Data Entry
1-3	EMD
5-6	Enter the total number of cards (maximum = 8) that are associated with the first 10 target types for this type of weapon.
8-9	Weapon type number
10-12	Weapon reliability (percent)
13-18	Enter CBU pattern length as a negative entry
19-24	Enter CBU pattern width as a positive entry

The supplementary cards permit the user to specify the expected percentage loss of various classes of resources when they are within the CBU bomblet pattern:

Card No.	Entry
13-18	Expected percentage loss at #1 type targets
19-24	Expected percentage loss at #2 type targets
o	
o	
o	
67-72	Expected percentage loss at #10 type targets

The first supplementary card is blank; the others are organized similarly to those for point-impact weapons. That is, the third through eighth cards apply to personnel, AGE and equipment, spare parts, munitions (or POL), TRAP, and building materials, respectively.

EQUI

The EQUIvalence cards are used to achieve consistency between the resources as defined in TSARINA and those defined in TSAR, when damage reports are prepared for transmission to TSAR. If the resources are defined identically in the data bases for both models, these cards are not required.

Resource equivalence data are entered using a specially formatted data string. The order of the data entered is: (1) the resource class, (2) the TSARINA resource type designator for which the equivalencies are defined, and (3) the TSAR designators of the resource types for which the percentage losses are to be equated to those of the designated TSARINA type. The numbers defining the resource classes are distinguished in the data string by addition of the number 5000.

If several TSAR resources are to be assigned the same TSARINA damage percentage, each of their numbers should immediately follow the number for the equivalent TSARINA resource. If necessary, the designator list can be continued on a subsequent card image when the eleven data fields are full; the list is terminated either by a null entry or by another resource class designator. However, the equivalence card cannot be used to equate a TSARINA resource designation to a type #0 resource to signify "all types not otherwise specified" for TSAR; to take advantage of that option, the TSARINA resource designation should itself be #0 (see p. 13). If the first entry following the TSARINA designator is -1, the TSARINA damage estimate for that resource is not reported to TSAR.

```
EQUI    5003   5   1   3   5   7   5003   6   2   4   6
EQUI      10  12   0
```

In this example, the first, third, fifth, and seventh types of aircraft spare parts, as defined for TSAR, are to be assigned the damage level assessed for the fifth type of spare part in TSARINA; similarly, the second, fourth, sixth, tenth, and twelfth TSAR spare parts are assigned the damage level assessed for the sixth type of spare part in TSARINA.

```
EQUI    5001   7   -1
```

In this case, estimates of casualties suffered by type #7 personnel are not to be reported to TSAR.

A final complication is introduced for differentiating between on-duty and off-duty personnel, and between assembled and unassembled munitions. TSAR personnel designators with values less than 1000 refer to on-duty personnel; off-duty personnel are specified by adding 1000 to their normal designator. Note that types 0 and 1000 refer, respectively, to all on-duty and off-duty personnel not otherwise specified. Unassembled munitions are designated by adding 100 to the nominal munition designation.

REDO

The REDO card is used to terminate the input for one case and initiate a new case with some or all of the previous inputs, as described earlier.

Columns	Data Entry
1-4	REDO
7-12	All targets will be retained unless the entry is unity (1); in that case a new set of targets and a new set of attacks are required.
13-18	The number of prior attacks to be retained when the targets are not changed. Each attack is numbered in the order in which it is entered; the attacks retained are selected from the top of that ordered list. All will be retained if there is no entry. Use a negative entry (-1) if none are to be retained.
19-24	An entry of unity (1) suppresses the input listings for targets and/or for attacks and weapons if no changes have been made in these data sets from the prior case.

END

An END card must be included at the end of all data entry cards.

Columns	Data Entry
1-3	END

Appendix B

DEFINITIONS OF VARIABLES AND ARRAYS USED IN TSARINA "COMMON" STATEMENTS

Key Variables

ACLOSS	The percentage of damaged aircraft that are not reparable.
ALLMC	Switch; internally set to unity when aircraft shelters are handled with the Monte Carlo mode.
BASE	The number of the airbase, in TSAR, at which the attack occurs.
CHANGE	Switch; set to unity between cases when the target data are to be changed.
DAY	The day, during the TSAR simulation, on which the air attack is presumed to occur.
HOUR	The hour, during the TSAR simulation, during which the air attack is presumed to occur.
INL	Distance along the runway the "minimum runway rectangle" is shifted.
INTSAR	Switch; set to unity when results are to be generated for TSAR.
INW	Lateral distance the minimum runway rectangle is shifted in checking for an adequate section.
ISAVE	Switch; set to unity if resource damage results are to be generated for the auxiliary FORMATER program.
ITRIAL	Number of the current trial.
KCBU	Switch; set to unity if any weapons are CBUs.
KPTI	Switch; set to unity if any weapons are the point-impact type.
KTEST	Index controlling variety of debugging printout options.
LAST	Switch; set to unity for last case.
LIST	Switch; when set to unity, target and/or attack input lists are suppressed when unchanged.

MCL	Minimum adequate length for required runway.
MCR	Switch; set to unity when runway availability is to be checked.
MCW	Minimum adequate width for required runway.
MINUTE	The minute, during the TSAR simulation, at which the air attack is presumed to occur.
MODE	Index controlling mode of operation.
MTT	Largest target-type number in the target array.
MXITEM	Maximum number of entries in the STOCKS array.
NA	Total number of weapon-delivery passes.
NAM	Maximum permissible number of weapon-delivery passes.
ND	Number of types of weapons in overall attack.
NHITD	Switch; set to unity when the expected-value mode is specified.
NJMEM	Number of weapon-delivery passes that require trajectory calculations.
NOAGE	Maximum number of entries in the AGE array; one greater than the size of the AGESTK array in TSAR.
NOEQUI	Maximum number of entries in the EQUIV array.
NOFAC	Maximum number of entries in the FACLTY array in TSAR.
NOMATL	Maximum number of entries in the MATERL array; one greater than the corresponding TSAR array.
NOMUN	Maximum number of entries in the AMMO array; 101 units greater than the MUNSTK array in TSAR.
NOPART	Maximum number of entries in the PARTS array; one greater than the corresponding TSAR array.
NOPEO	Dimension of PEOPLE array in TSARINA; equals (2*NOPEOP +2).
NOPEOP	Maximum number of entries in the PEOPLE array in TSAR.
NOPOL	Maximum number of entries in the POL array.

NOTRAP	Maximum number of entries in the TRAP array; one greater than the corresponding TSAR array.
NPLOT	Switch; set to 1 or 2 if runway impact plots are desired.
NPRINT	Index controlling results output.
NREDO	Switch; set to unity if an additional case is specified.
NREP	Switch; set to unity when repair requirements are to be assessed.
NSAVE	The number of weapon-delivery passes saved from one case to be used in the next case.
	Number of targets to be retained for a subsequent case.
	Number of weapon-delivery passes to be retained for a subsequent case.
NSM	Total number of aircraft shelters.
NST	Maximum number of targets for which hits can be stored.
NSTAT	Cumulative number of trials in which the minimum runway was available.
NT	Total number of targets entered using the TGT cards.
NT2	Total number of targets entered using the TGT2 cards.
NTM	Maximum permissible number of targets.
NTRIAL	Total number of trials specified.
PDAM	Switch; position controls output formats for trial-to-trial damage summaries.
PUNCH	When greater than zero, output for TSAR is card-punched.
RAMPS	Target-type chosen to designate aircraft parking aprons and ramps.
REPAIR	The minimum number of crater repairs required to clear the minimum area for flight operations.
SHELT	Target-type chosen to designate aircraft shelters.
TSAR	When greater than zero, output for TSAR is stored on disk.
TXWYS	Target-type chosen to designate taxiways.

KEY ARRAYS

All arrays listed in labeled Common in TSARINA are defined below. The first seven arrays store data pertaining to AGE and equipment, munitions, building materials, aircraft spare parts, personnel, POL, and TRAP, respectively. The definitions shown below the array names are the same for all of these arrays.

AGE(I,J)

AMMO(I,J)

MATERL(I,J)

PARTS(I,J)

PEOPLE(I,J)

POL(I,J)

TRAP(I,J)

I =		Resource subcategory
J = 1		Pointer to the location in the STOCKS array, where the first quantity of this resource is stored.
2		Cumulative losses at all targets where this resource is stored.
3		Square of the cumulative losses.
4		Pointer to the location in the EQUIV array, where the first equivalent TSAR resource category designations are stored.

ATT(I,J) Storage array for weapon-delivery data.

I	Weapon-delivery pass number; numbered internally in order of entry.
J = 1	Heading (deg).
2	X-coordinate of desired mean point of impact.
3	Y-coordinate of DMPI.
4	Range error probable of DMPI.
5	Deflection error probable of DMPI.

6	Dispersion in range (ground plane).
7	Number of weapons released in pass.
8	Length of stick (in ground plane).
9	Weapon type.
10	Dispersion in deflection.
11	Probability attacker arrives at target.
CBUHT(J,K)	Impact coordinates of the centroid of the Jth CBU pattern.
K = 1	X-coordinate.
2	Y-coordinate.
COV(L)	Fraction of target L covered by one or more CBU patterns.
COV2(I,J)	
J = 1	Expected number of weapons that impact within R1 feet of aircraft shelter "I".
2	Expected number of weapons that impact within R2 feet of aircraft shelter "I".
EMD(I,J,K)	Weapon effectiveness data.
I	Weapon type.
J	Target type.
K = 1	Effective miss distance R1 for a near miss.
2	Effective damage radius R2 for a near miss.
3	Personnel damage factor for a near miss.
4	Equipment damage factor for a near miss.
5	Aircraft spare parts damage factor for a near miss.
6	Munitions damage factor for a near miss.
7	TRAP damage factor for a near miss.
8	Building material damage factor for a near miss.

9	Coded Flag defining the criteria for assessing resource damage.
10	Effective miss distance R1 for a direct hit.
11	Effective damage radius R2 for a direct hit.
12	Personnel damage factor for a direct hit.
13	Equipment damage factor for a direct hit.
14	Aircraft spare parts damage factor for a direct hit.
15	Munitions damage factor for a direct hit.
16	TRAP damage factor for a direct hit.
17	Building material damage factor for a direct hit.
EQUIV(NOEQUI)	Used to store the resource designators to be used for reporting damage to TSAR.
FACLTY(I)	Used to store the TSAR facility number for those structures whose damage is to be reported to TSAR.
HIT(I,J,K)	Storage array for hit locations on specified targets.
I	Ith of those targets for which hit data are to be stored.
J = 1	X-coordinate.
2	Y-coordinate.
3	Weapon type.
K	Number of hits on the Ith target.
HITR(I,J,K)	Storage array for hit locations on type #1 targets (i.e., runways and taxiways).
I,J,K	See HIT(I,J,K).
IR(N)	Switch; set to unity if the Nth weapon-delivery attacker fails to reach target.
IZ(I)	Designates the zone for each target (see subroutine TGTZON).

IZONE(K,J)	Denotes which of the ordered targets fall in the Kth target zone.
J = 1	Lowest numbered target in the Kth zone.
2	Highest numbered target in the Kth zone.
MHIT(K)	Target number of the Kth target for which hit location data are to be stored.
MSTAT(J)	Storage array for accumulating trial results of runway availability tests.
J = 1	Minimum number of repairs required to open a minimum runway.
2	Square of J = 1, above.
3-8	Not used.
MTYPE(I)	Index that specifies whether or not supplementary data are to follow the EMD card for weapon type I.
NAME(I,4)	Stores either a two-word or four-word alphanumeric name for each target.
I	Target number
NCBU(L)	Number of CBU weapon patterns that cover all or part of target L.
NHIT(L)	Number of hits on target L; by both point-impact and CBU weapons.
NRW(I)	Target number of the Ith runway entered.
OHIT(I)	Counts near misses for each target.
P(L,K)	Damage estimates for targets handled with the Monte Carlo mode.
K = 1	Expected fraction of target L that is within the radius R1 of point-impact weapons.
2	Expected fraction of target L that is within the radius R2 of point-impact weapons.
3	Fraction of personnel casualties expected at target L.

4	Fraction of equipment losses expected at target L.
5	Fraction of spare parts losses expected at target L.
6	Fraction of munitions (or POL) losses expected at target L.
7	Fraction of TRAP losses expected at target L.
8	Fraction of building material losses expected at target L.
P2(L,K)	Damage estimates for aircraft shelters.
K = 1	Expected fraction of target L within radius R1 of the weapon impacts.
2	Expected fraction of target L within radius R2 of the weapon impacts.
3	Probability that a sheltered aircraft is damaged when the shelter door is closed.
4	Probability that the shelter is killed.
5	Probability that a sheltered aircraft is damaged when the shelter door is open.
6	Fraction of personnel casualties expected at target L.
7	Fraction of equipment losses expected at target L.
8	Fraction of spare parts losses expected at target L.
SHEL(N)	The TSARINA-generated target number for the Nth shelter.
STAT(L,J)	Storage array for accumulating trial results.
L	Target number.
J = 1	Number of hits by point-impact weapons.
2	Square of J = 1, above.
3	Trials with at least one hit.
4	Fractional coverage by CBU weapons.
5	Square of J = 4, above.

	6	Fractional target coverage within radius R1 of point-impact weapons.
	7	Fractional target coverage within radius R2 of point-impact weapons.
	8	Fractional personnel casualties.
	9	Fractional equipment losses.
	10	Fractional spare parts losses.
	11	Fractional munitions losses.
	12	Fractional TRAP losses.
	13	Fractional building material losses.
STAT2(I,J)		Storage array for accumulating trial results for targets of a given type.
I		Target type.
J = 1		Fraction of the targets of type I that received at least one hit.
	2	Square of J = 1, above.
STOCKS(I,J)		Resource storage location infomation.
J = 1		Target number at which resource is located.
	2	The percent of the resource stored in this location (in tenths of percent).
	3	Pointer to next target with the same type of resource.
TGT(L,J)		Storage array for target data.
L		Target number; numbered internally in order of entry.
J = 1		X-coordinate of westernmost corner (#1).
	2	Y-coordinate of corner #1.
	3	X-coordinate of corner #2.
	4	Y-coordinate of corner #2.
	5	X-coordinate of corner #3.

6	Y-coordinate of corner #3.
7	X-coordinate of corner #4.
8	Y-coordinate of corner #4.
9	Heading of northeast target leg.
10	Target type.
11	Switch; hits stored when reset to unity.
12	Dimension of northeast target leg.
13	Dimension of southeast target leg.
14	Facility number of target.
TGT2(L,J)	Storage array for aircraft shelter data.
L	Target number; numbered internally in order of entry.
J = 1	X-coordinate of westernmost corner (#1).
2	Y-coordinate of corner #1.
3	X-coordinate of corner #2.
4	Y-coordinate of corner #2.
5	X-coordinate of corner #3.
6	Y-coordinate of corner #3.
7	X-coordinate of corner #4.
8	Y-coordinate of corner #4.
9	Heading of northeast target leg.
10	Target type.
11	Switch; hits stored when reset to unity.
12	Dimension of northeast target leg.
13	Dimension of southeast target leg.
14	Facility number of target.

TO(I,J)	Target order array in which targets are ordered according to increasing values of the sum of the coordinates of the western corner.
I	Ith target in the ordered array.
J = 1	Value of (X+Y) for the Ith ordered target.
2	Number of the target as initially entered.
WPNREL(I)	Reliability of weapon type I.

Appendix C

TSARINA SOURCE CODE AND COMMENTS

```
C     MAIN - TSARINA - AIRBASE DAMAGE ASSESSMENT INPUTS FOR TSAR.
            IMPLICIT INTEGER *2 (A-Z)
            INTEGER *4   MSTAT, NBASE1, NBASE2
            REAL *4  STAT
C      THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
//          DD     DSN=*.STEP1.COMMON,DISP=SHR
//          DD     *,DCB=BLKSIZE=800
C
C     *******  "BASIC" COMMON IS STORED IN TSARIN.COMMON  ************
C                FILE AS "SAVE TSARIN.COMMON REP CARD (10)"
C
C           INTEGER*4   NAME, EMD
C           REAL   *4   P, P2, COV, COV2, WPNREL
C
C         COMMON /BASIC/ NAME(500,4),P(500,8),P2(100,8),COV(500),
C      X      EMD(10,20,17), MTYPE(10), COV2(100,2), WPNREL(10),
C      X      IZ(500), NT, TGT(500,14), NT2, TGT2(100,13), SHEL(100),
C      X      NA, ATT(50,11), ND, NTM, NSM, NAM, MTT, NST, ALLMC,
C      X      MODE, KTEST, NPRINT, PDAM, NREDO, MCR, MCL, MCW, NRW(5),
C      X      NAMES, LAST, BASE, DAY, HOUR, MINUTE, ACLOSS,
C      X      INTSAR,REPAIR,FACLTY(500),NOFAC,SPEC,SHELT,TXWYS,RAMPS,
C      X      TO(500,2), IZONE(100,2), NHIT(500), OHIT(500), MHIT(20),
C      X      HIT(20,3,25), HITR(5,3,250), NCBU(500)
C
C     ***************************************************************
C       THE LINEAR ARRAY 'ZO , AND THE 2 ZEROING LOOPS, BELOW, MUST
C       BE REDIMENSIONED IF ANY PART OF THE COMMON 'STORES' IS CHANGED.
C           **** NOTE THAT ALL ENTRIES ARE *4 EXCEPT 'STOCKS'
C     ***************************************************************
          INTEGER *4   MXITEM,NOPEO,NOAGE,NOPART,NOMUN,NOTRAP,NOMATL,NOPOL
          INTEGER *4   PEOPLE,AGE,PARTS,AMMO,TRAP,MATERL,POL,NOEQUI,NOPECP
          COMMON /STORES/ MXITEM,NOPEO,NOAGE,NOPART,NOMUN,NOTRAP,NOMATL,
       X  NOPOL,NOECUI,NOPEOP,PEOPLE(202,4),AGE(101,4),PARTS(401,4),
       X  AMMO(126,4),TRAP(26,4),MATERL(26,4),POL(10,4),STOCKS(1000,3),
       X  EQUIV(400)
C     ***************************************************************
          COMMON /STATS / STAT(500,17),MSTAT(8),NTRIAL,ITRIAL,NSTAT
          COMMON /CONTRL/ NREP,NPLOT,INW,INL,CHANGE,NSAVE,LIST,NJMEM
          COMMON /OUTPUT/ TSAR, PUNCH, NBASE1, NBASE2
C
C      THE CBU CODE IS LIMITED TO 200 CLUSTER BOMBLET CONTAINER IMPACTS
C
          COMMON /CBUHIT/ CBUHT(200,2), IR(50), KCBU, KPTI
C     ***************************************************************
          DIMENSION ZO(7402)
          EQUIVALENCE  (ZO(1), MXITEM)
C     ***************************************************************
          NTM = 500
C    *****  NTM IS THE MAXIMUM NUMBER OF TARGETS  *****
C             ARRAYS:  TGT, TO, NHIT, OHIT, P, COV, NAME, NCBU, STAT,
C                      FACLTY, IZ
          NT2M = 100
C    *****  NT2M IS THE MAXIMUM NUMBER OF SHELTERS, WHEN THEY ARE AGGRAGATED
C             ARRAYS:  TGT2, P2, COV2, SHEL   (SEE ALSO EHIT1 IN EXPHIT)
          NAM = 50
C    *****  NAM IS THE MAXIMUM NUMBER OF ATTACKS
C             ARRAYS:   ATT, IR
          NST = 20
C    *****  NST IS THE NUMBER OF TARGETS FOR WHICH HITS CAN BE STORED.
C             ARRAYS:  HIT, MHIT
C    *****  TO CHANGE ANY OF THE PRECEDING DIMENSIONS, MAKE THE APPROPRIATE
C    *****  CHANGES IN THE ARRAYS AND THEN CHANGE THE LIMITING VALUE.
C    ******************************************************************
C
C      THE OUTPUTS FOR TSAR MAY BE PRINTED, PUNCHED ON CARDS AND FILED
C      DIRECTLY ON DISK FOR PRODUCTION RUNS.  THE 2ND AND 3RD OPTIONS
C      ARE CONTROLLED BY THE VARIABLES "PUNCH" AND "TSAR", RESPECTIVELY,
C      AND ARE EXERCISED WHEN THOSE VARIABLES ARE INITIALIZED TO UNITY.
C
C    ******************************************************************
C
          DO 1   I = 1, 10556
     1    ZO(I) = 0
C
```

```
C     THE NUMBER OF FACILITIES AND THE DIMENSIONS OF THE ARRAYS
C     IN THE "STORES" COMMON MUST BE ENTERED HERE.  THE DIMENSIONS
C     SHOULD BE ONE LARGER THAN THE ARRAYS USED IN THE COMPANION
C     VERSION OF THE TSAR SIMULATION MODEL, EXCEPT FOR MUNITIONS
C     AND PERSONNEL.   FOR MUNITIONS THE DIMENSION SHOULD BE LARGER
C     BY +101 AND FOR PERSONNEL IT SHOULD BE 2*NOPEOP + 2, WHERE
C     NOPEOP IS THE NUMBER OF PERSONNEL TYPES AS USED IN TSAR.
C
            NOFAC = 60
C
            MXITEM = 1000
            NOPEO  =  202
            NOAGE  =  101
            NOPART =  401
            NUMUN  =  126
            NOTRAP =   26
            NOMATL =   26
            NOPOL  =   10
            NOEQUI =  400
            NOPEOP =  (NOPEO - 2)/2
C
          NT = 0
          NT2 = 0
          NA = 0
          NP = 0
          MCR = 0
          CASE = 0
          MWPN = 0
          NTRIAL = 1
          KTEST = 0
          NREDY = 0
          CHANGE = 0
          NEWTGT = 1
          DO 2   I = 1,10
          MTYPE(I) = 0
          WPNRE(I) = 1.0
          DO 2   J = 1,20
          DO 2   K = 1,7
     2    EMO(I,J,K) = 0
          DO 3   I = 1, NIM
     3    FACTY(I) = 0
     4    ITRIAL = 0
          FLAG = 0
C
C     REPLACE RESOURCE LOCATION DESCRIPTORS AS REQUIRED
C
          IF (CHANGE .GE. 1)  GO TO 1020
          NT  = 0
          NT2 = 0
          CHANGE = 0
          FLAG = 1
          NEWTGT = 1
          DO 1010  I = 21, 10556
   1010   Z0(I) = 0
   1020   CONTINUE
C
C     *****   BEGIN NEW CASE   *****
C
          CASE = CASE + 1
          DO 5   I = 1, 5
     5    NRW(I) = 0
C
          CALL INPUT(CASE,MWPN,NEWTGT,NTSAF,NPUNCH)
C
          IF (NTRIAL .LT. 2)   GO TO 25
          IF ((CASE .EQ. 1).OR.(FLAG .EQ. 1))  GO TO 14
          DO 12  J = 2,3
          DO 6  I = 1,NOPEO
     6    PEOPLE(I,J) = 0
          DO 7  I = 1,NOAGE
     7    AGE(I,J) = 0
          DO 8  I = 1,NOPART
     8    PARTS(I,J) = 0
          DO 9  I = 1,NUMUN
     9    AMMO(I,J) = 0
          DO 10  I = 1,NOTRAP
```

```
 10   TRAP(I,J) = 0
      DO 11  I = 1,NOMATL
 11   MATERL(I,J) = 0
      DO 12  I = 1,NOPOL
 12   POL(I,J) = 0
 14   CONTINUE
      NSTAT = 0
      DO 15   I = 1,NT
      DO 15   N = 1,13
 15   STAT(I,N) = 0.0
 25   DO 30   I = 1, NST
 30   MHIT(I) = 0
 40   CONTINUE
      DO 45   N = 1,8
 45   MSTAT(N) = 0
      CALL TGTDIM
      DO 50   I = 1,NT2M
      DO 50   J = 1,8
 50   P2(I,J) = 0.0
      IF (NT2 .EQ. 0)   GO TO 60
      DO 55   I = 1,NT2
      DO 55  N = 1,2
 55   COV2(I,N) = 0.0
C
      CALL EXPHIT
C
     IF (NT .EQ. 0)  GO TO 200
 60   IF (KPTI .EQ. 0)   GO TO 100
C
      CALL TGTORD
C
      CALL TGTZON
C
 100  CONTINUE
C  *****   BEGIN NEW TRIAL   *****
     ITRIAL = ITRIAL + 1
      IF (ITRIAL .GT. NTSAR)   TSAR = 0
      IF (ITRIAL .GT. NPUNCH)  PUNCH = 0
      DO 105  I1 = 1, NST
      DO 105  I2 = 1,3
      DO 105  I3 = 1, 25
 105  HIT(I1,I2,I3) = 0
      DO 110 I = 1,NT
      COV(I) = 0.0
      DO 108  N = 1,8
 108  P(I,N) = 0.0
      NCBU(I) = 0
      OHIT(I) = 0
 110  NHIT(I)=0
      IF (NSM .EQ. 0)  GO TO 114
      DO 112  I = 1, NT2M
      DO 112  N = 1, 8
 112  P2(I,N) = 0.0
 114  CONTINUE
      DO 115  I1 = 1, 5
      DO 115  I2 = 1, 3
      DO 115  I3 = 1, 250
 115  HITR(I1,I2,I3) = 0
      CALL BOMB
      IF (KCBU .EQ. 1)   CALL CBU
      REPAIR = 0
      IF (MCR .NE. 0)   CALL CHECKR
      IF (NTRIAL .LT. 2)   GO TO 170
      DO 140   I = 1, NT
      AID = NHIT(I) - NCBU(I)
      STAT(I,1) = STAT(I,1) + AID
      STAT(I,2) = STAT(I,2) + AID*AID
      IF ( AID .GT. 0.0)   STAT(I,3) = STAT(I,3) + 1.
      STAT(I,4) = STAT(I,4) + COV(I)
      STAT(I,5) = STAT(I,5) + COV(I)*COV(I)
      DO 140  J = 1,8
      STAT(I,J+5) = STAT(I,J+5) + P(I,J)
 140  CONTINUE
 170  CONTINUE
      IF (NPRINT .GT. 1)   GO TO 180
      CALL PRINT
```

```
      180   CONTINUE
            IF (NPRINT .NE. 5)   GO TO 190
            DO 185   L = 1,NT
      185   WRITE(6,1001)  ITRIAL, L, NHIT(L)
      190   CONTINUE
            IF (INTSAR .EQ. 1)  CALL DAMAGE
            IF (ITRIAL .LT. NTPIAL)   GO TO 100
            IF (NTRIAL .GT. 1)  CALL STATIS
      200   IF ((NT .EQ. 0).AND.(INTSAR .EQ. 1))  CALL DAMAGE
            IF (NREDO .EQ. 1)   GO TO 4
            STOP
     1001   FORMAT(' ','TRIAL',I4,'    TGT',I4,'    HITS',I4)
            END
```

```
      SUBROUTINE INPUT(CASE,MWPN,NEWTGT,NTSAR,NPUNCH)
        IMPLICIT INTEGER *2 (A-Z)
      INTEGER *4   DATA,LABEL,AN,WORDS,NBASE1,NBASE2,AID
      INTEGER *4   DIM,DIM2,MSTAT
      INTEGER *4   MXITEM,NOPEO,NOAGE,NOPART,NOMUN,NOTRAP,NOMATL,NOPOL
      INTEGER *4   PEOPLE,AGE,PARTS,AMMO,TRAP,MATERL,POL,NOEQUI,NOPEOP
      REAL    *4   STAT
C   THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
//      DD    DSN=*.STEP1.COMMON,DISP=SHR
//      DD    *,DCB=BLKSIZE=800
C
      COMMON /STATS / STAT(500,17),MSTAT(8),NTRIAL,ITRIAL,NSTAT
      COMMON /CONTRL/ NREP,NPLOT,INW,INL,CHANGE,NSAVE,LIST,NJMEM
      COMMON /CBUHIT/ CBUHT(200,2), IR(50), KCBU, KPTI
      COMMON /OUTPUT/ TSAR, PUNCH, NBASE1, NBASE2
C
C   THE DATA STORAGE ARRAYS (AND DIMENSIONS) FOR THE RESOURCE
C   STORAGE LOCATION DATA ARE FILED IN LABELLED COMMON STORES.
C
C         SEE THE MAIN ROUTINE FOR DIMENSIONING INSTRUCTIONS
C
      COMMON /STORES/ MXITEM,NOPEO,NOAGE,NOPART,NOMUN,NOTRAP,NOMATL,
     X NOPOL,NOEQUI,NOPEOP,PEOPLE(202,4),AGE(101,4),PARTS(401,4),
     X AMMO(126,4),TRAP(26,4),MATERL(26,4),POL(10,4),STOCKS(1000,3),
     X EQUIV(400)
C
        COMMON / LISTER /  ISAVE                                        ***
C                                                                       ***
      DIMENSION LABEL(10), DIM(500,2), DIM2(100,2), DATA(11), WORDS(4),
     X          CARDS(10)
      DATA LABEL /'TGT ','TGT2','ATT ','ATT2','EMD ','EQUI','CONT',
     X    'DATA','REDO','END ' /
C
C
       NS = 0
       XMIN = 0
       YMIN = 0
      NJMEM = 0
      LAST  = 0
      LIST1 = 0
      LIST2 = 0
      ITEM  = 0
      FLAG = 0
      IF (CASE .EQ. 1)  NAMES = 2
      NRMAX = 0
        NAO = 1
      IF (NREDO .EQ. 0)    GO TO 2
C    SEE NOTE AT LABEL '48'
      IF (NEWTGT .EQ. 0)  LIST1 = LIST
      IF (NSAVE .GT. 0)    NA = NSAVE
      IF (NSAVE .LT. 0)    NA = 0
      NAO = NA + 1
      IF (NSAVE .EQ. 0)    LIST2 = LIST
      NSAVE = 0
    2 CONTINUE
      NREDO = 0
    6 READ (5,101) AN, NTYPE, (DATA(I), I=1,11), (WORDS(I),I=1,NAMES)
       IF(KTEST.GT.2)WRITE(6,1101)AN,NTYPE,(DATA(I),I=1,11),
     X                          (WORDS(I),I=1,NAMES)
C    NO ENTRY IS REQUIRED IN COLUMNS 5 AND 6.  IF AN INTEGER IS FOUND
C    ON AN ATTACK CARD, THE ATTACK WILL BE REPEATED SO THAT THERE WILL
C    BE THAT TOTAL NUMBER OF ATTACKS WITH THE STATED CHARACTERISTICS.
C        (ONE ATTACK IS ASSUMED IF THERE IS NO ENTRY.)
C    THE ENTRY IN COL 6 ON THE EMD CARDS AND COLUMNS 67-70 OF
C    THE TGT CARDS  CONTROL THE NUMBER OF CARDS EXPECTED.
C
      IF (AN .EQ. LABEL(1)) GO TO 10
      IF (AN .EQ. LABEL(2)) GO TO 15
      IF (AN .EQ. LABEL(3)) GO TO 20
      IF (AN .EQ. LABEL(4)) GO TO 26
      IF (AN .EQ. LABEL(5)) GO TO 30
      IF (AN .EQ. LABEL(6)) GO TO 36
      IF (AN .EQ. LABEL(7)) GO TO 40
      IF (AN .EQ. LABEL(8)) GO TO 45
      IF (AN .EQ. LABEL(9)) GO TO 48
      IF (AN .EQ. LABEL(10)) GO TO 50
```

```
          WRITE(6,1102) AN,TYPE,(DATA(I),I=1,11),(WORDS(I),I=1,NAMES)
          GO TO 6
   10     NT = NT + 1
          IF ((NT .EQ. 1).AND.(NPRINT .LE. 1))  WRITE(6,121)
          IF (NEWTGT .EQ. 0)  GO TO 170
          LIST1 = 0
          IF (NT .GT. NTM)   GO TO 120
          DO 11 I = 1,2
   11     DIM(NT,I) = DATA (I)
            IF (DATA(1) .LT. XMIN)   XMIN = DATA(1)
            IF (DATA(2) .LT. YMIN)   YMIN = DATA(2)
          DO 13 I = 3,4
   13     TGT(NT,I+9) = DATA(I)
          DO 131 I = 5,7
  131     TGT(NT,I + 4) = DATA (I)
          TGT(NT,14) = DATA(8)
          IF (DATA(6) .NE. SHELT)  GO TO 133
          NS = NS + 1
          TGT(NT,14) = 1000 + NS
          SHEL(NS) = NT
  133     IF ((DATA(8).GT.0).AND.(DATA(8).LT.1000))  FACLTY(DATA(8)) = NT
          DO 14  I = 1,NAMES
   14     NAME(NT,I) = WORDS(I)
          IF ((DATA(9)+DATA(11)) .EQ. 0)  GO TO 142
          IF (NPRINT .LE. 1)  WRITE(6,122)  NT, (NAME(NT,L),I=1,4),
        X                          DATA(1), DATA(2)
          ITEM = ITEM + 1
          IF (ITEM .GT. MXITEM)  GO TO 160
          NCARDS = DATA(11)/100
          IF (NCARDS .EQ. 0)  GO TO 141
C
C     SUBROUTINE STORE ORGANIZES RESOURCE DATA FOR
C     FACILITIES WITH SEVERAL ITEMS.
C
            CALL STORE( ITEM, NCARDS )
C
          GO TO 142
C
C     WITH ONLY ONE CLASS AND ONE TYPE (OR ALL TYPES OF A CLASS),
C     ENTRY FILE IN SUBROUTINE STORE IS USED.
C
  141     STOCKS(ITEM,1) = NT
          STOCKS(ITEM,2) = 1000
C
            CALL FILE(ITEM,DATA(9),DATA(10) )
C
  142     CONTINUE
          GO TO 6
   15     NT2 = NT2 + 1
          IF (NEWTGT .EQ. 0)  GO TO 170
          IF (NT2 .GT. 100)   GO TO 120
          DO 16 I = 1,2
   16     DIM2(NT2,I) = DATA (I)
          IF (DATA(1) .LT. XMIN)  XMIN = DATA(1)
          IF (DATA(2) .LT. YMIN)  YMIN = DATA(2)
          DO 17 I = 3,4
   17     TGT2(NT2,I+9) = DATA(I)
          DO 18 I = 5,7
   18     TGT2(NT2,I + 4) = DATA (I)
          GO TO 6
   20     NA = NA + 1
          LIST2 = 0
           IF ( NA .GT. NAM)   GO TO 130
          DO 22   I = 1,6
   22     ATT(NA,I) = DATA(I)
          ATT(NA,10) = DATA(7)
          IF (DATA(7) .EQ. 0)    ATT(NA,10) = ATT(NA,6)
          IF (DATA(11) .EQ. 0)    DATA(11) = 100
          ATT(NA,11) = DATA(11)
          DO 24   I = 7,9
   24     ATT(NA,I) = DATA(I+1)
          NTYPE = MAX0((NTYPE-1),0)
         IF (NTYPE .EQ. 0)   GO TO 6
         GO TO 20
   26    NA = NA + 1
          LIST2 = 0
```

```
        IF ( NA .GT. NAM)   GO TO 130
        IF (DATA(11) .EQ. 0)     DATA(11) = 1.0
        CALL JMEMO(NJMEM,DATA,NA,KTEST)
        DO 27   I = 1,11
   27  ATT(NA,I) = DATA(I)
   28  NTYPE = MAX0((NTYPE-1),0)
       IF (NTYPE .EQ. 0)   GO TO 6
       NA = NA + 1
        DO 29   I = 1,11
   29  ATT(NA,I) = ATT((NA-1),I)
       GO TO 28
   30   ND = ND + 1
        NR = NTYPE
        IF (NR .EQ. 0)  NR = 1
        IF (NR .GT. NRMAX) NRMAX = NR
        CBU = 0
        IF (DATA(2) .LT. 0)  CBU = 1
        M = DATA(1)/1000
        IF (M .GT. 10)  GO TO 140
        IF (M .GT. MWPN)  MWPN = M
        AID = DATA(1) - 1000*M
        IF (AID .EQ. 0)  AID = 100
        WPNBEL(M) = AID/100.
        MTYPE(M) = NR
        CARDS(M) = TT
        DO 32  N = 2,11
   32   EMD(M,N-1,1) = DATA(N)
        DO 34  NC = 2, NR
   34   READ(5,114)  (EMD(M,N,NC), N = 1,10 )
        IF (TT .EQ. 1)  GO TO 6
        NF = CBU + 1
        DO 35  NC = NF, NR
   35   READ(5,114)  (EMD(M,N,NC), N = 11,20)
        GO TO 6
C
C     STORE RESOURCE EQUIVALENCE DATA
C
   36     DO 39  I = 1,11
          IF ((DATA(I) .EQ. 0).AND.(FLAG .GT. 0))  GO TO 39
          IF (DATA(I) .GT. 5000)  GO TO 38
          IF (TYPE .EQ. 0)  GO TO 37
C
        CALL SAVE(CLASS, TYPE, DATA(I))
C
          FLAG = FLAG + 1
          GO TO 39
   37     TYPE = DATA(I)
          GO TO 39
   38     CLASS = DATA(I) - 5000
          TYPE = 0
          FLAG = 0
   39     CONTINUE
          GO TO 6
C
C    TSAR INPUTS ARE PREPARED AND LISTED IF INTSAR IS SET TO UNITY.
C
   40   INTSAR = NTYPE
        MODE = DATA(1)/1000
        TTGT = DATA(1) - 1000*MODE
        IF (TTGT .LT. 0)  TTGT = -TTGT
        IF (TTGT .LE. 10)  TT = 1
        IF (TTGT .GT. 10)  TT = 2
        AID = DATA(2)
        NTRIAL = AID/1000
        AID = AID - 1000*NTRIAL
        IF (NTRIAL .LT. 2)   NTRIAL = 1
        IF (AID .GT. 0)  NAMES = 4
        NPRINT = DATA(3)/1000
          PDAM = DATA(3) - 1000*NPRINT
          IF (PDAM .LT. 0)  PDAM = - PDAM
        KTEST  = DATA(4)
          CALL SAVER(0,KTEST,MAXLOC)
        MCR = 0
        IF (DATA(5) .GT. 0)   MCR = 1
        MCL = DATA(5)
        MCW = DATA(6)
```

```
          NREP = DATA(7)/1000
          NPLOT= DATA(7) - 1000*NREP
         INW = DATA(8)
         INL = DATA(9)
          IF (INW .EQ. 0)   INW = 5
         IF (INL .EQ. 0)   INL = 250
          NBASE1 = WORDS(1)
          NBASE2 = WORDS(2)
         GO TO 6
C
C   THE DATA CARD ONLY NEED BE USED WHEN OUTPUT IS TO BE USED WITH TSAR
C
C           IF THERE IS AN ENTRY IN COLUMNS 5-6 FOR NTYPE (ISAVE),
C           THE LOSSES FOR EACH SUBCATEGORY OF RESOURCES ARE
C           PUNCHED IN CARD FORMAT SO AS TO BE READ DIRECTLY BY
C           THE "FORMATER" AUXILIARY PROGRAM AND PROCESSED TO A
C           FORM THAT IS USABLE DIRECTLY IN THE "PREPARE.RESULTS"
C           AUXILIARY PROGRAM.
C
C                  ALL RELATED CARDS ARE IDENTIFIED
C                  BY A  ***  IN COLUMNS 73-75.
C
   45   CONTINUE
        ISAVE  = NTYPE                                                    ***
        TSAR   = DATA(1)
        NTSAR  = TSAR
        IF (NTSAR .GT. 0)  TSAR  = 1
        PUNCH  = DATA(2)
        NPUNCH = PUNCH
        IF (NPUNCH .GT. 0)  PUNCH = 1
        BASE   = DATA(3)
        DAY    = DATA(4)
        HOUR   = DATA(5)
        MINUTE = DATA(6)
        SHELT  = DATA(7)
        TXWYS  = DATA(8)
        RAMPS  = DATA(9)
C
C   ACLOSS IS THE EXPECTED LOSS-TO-DAMAGE PERCENTAGE FOR AIRCRAFT.
C            (SEE SUBROUTINE DAMAGE FOR APPLICATION)
C
        ACLOSS = DATA(10)
C                                                                         ***
                              IF (ISAVE .EQ. 0)  GO TO 6                  ***
                              N99 = 99                                    ***
                              WRITE(7,1111) N99,BASE,ISAVE                ***
 1111                         FORMAT(I3,2X,I2,5X,I2)                      ***
C                                                                         ***
        GO TO 6
   48   NREDO = 1
C
C     IF THE FIRST DATA ENTRY ON THE REDO CARD (COL 12) IS "1",
C     ALL THE TARGETS ARE TO BE CHANGED FOR THE NEXT CASE.
C
        CHANGE = DATA(1)
C
C     THE SECOND DATA ENTRY ON THE REDO CARD (COL 18) MAY BE USED
C     TO SPECIFY THE NUMBER OF PRIOR ATTACKS TO BE INCLUDED IN
C     THE NEW CALCULATION.  THAT NUMBER WILL BE SELECTED IN
C     RANK ORDER FROM THOSE INPUT PREVIOUSLY; IN NO CASE MAY THE
C     NUMBER BE LARGER THAN THE NUMBER AVAILABLE.   IF NO NUMBER
C     IS ENTERED, ALL PRIOR ATTACKS WILL BE INCLUDED.  IF A
C     NEGATIVE NUMBER IS ENTERED (EG  -1), NONE OF THE PRIOR
C     ATTACKS WILL BE TREATED.  NEW ATTACKS ARE REQUIRED
C     IF NEW TARGETS ARE ENTERED.
C
        NSAVE = DATA(2)
        IF (CHANGE .GT. 0)  NSAVE = -1
C
C     IF THE THIRD ENTRY (COL 24) IS SET TO UNITY, THE TARGET LIST
C     AND/OR THE ATTACK/WEAPON LISTS WILL BE SUPPRESSED IF THE
C     LATTER HAVE NOT BEEN CHANGED.
C
        LIST = DATA(3)
   50   CONTINUE
C
```

```
      C   PLACE A "-1" IN THE TENTH ROW OF EACH COLUMN FOR THOSE TARGET
      C   TYPES FOR WHICH THE EFFECTIVENESS WITH INTERNAL AND EXTERNAL HITS
      C   IS NOT TO BE DISTINGUISHED.
      C
            LMT = 10*TT
            DO 54  M = 1,10
            IF (MTYPE(M) .EQ. 0)  GO TO 54
            DO 54  N = 1,LMT
            IF (N .EQ. SHELT)  GO TO 54
            IF (NRMAX .LE. 9)  GO TO 53
            DO 51  I = 1,NRMAX
            ROW = NRMAX - I + 1
            IF (EMD(M,N,ROW) .NE. 0)  GO TO 52
            IF (ROW .EQ. 9)  GO TO 53
         51 CONTINUE
         52 IF (ROW .EQ. 9)  EMD(M,N,10) = -1
            GO TO 54
         53 EMD(M,N,10) = -1
         54 CONTINUE
      C
      C  NOTE WHEN SHELTERS ARE TO BE HANDLED WITH THE MONTE
      C  CARLO MODE RATHER THAN THE EXPECTED VALUE MODE.
      C
            ALLMC = 0
            IF ((SHELT .GT. 0).AND.(NT2 .EQ. 0))  ALLMC = 1
            NSM = NS
      C
              CALL SAVER(1,KTEST,MAXLOC)
      C
              WRITE(6,128)  ITEM, MAXLOC
      C
      C  PRINT OUTPUT HEADING BLOCK
      C
            IF (NREDO .EQ. 0)  LAST = 1
            IF (MODE .EQ. 0)   CALL RSTART(7)
            WRITE (6,111) NTRIAL, NPRINT, PDAM, MODE, MCL, INL, MCW, INW,
          X   NREP, NPLOT, KTEST, CASE, BASE, DAY, HOUR, MINUTE
            IF ((NBASE1 .EQ. 0) .AND. (NBASE2 .EQ. 0))  GO TO 55
            WRITE (6,100)   NBASE1, NBASE2
        55  CONTINUE
              IF (NPRINT .GT. 0)  WRITE(6,129)  NT, NA
              IF (NPRINT .LE. 0)  WRITE(6,102)
              IF (NEWTGT .EQ. 0)  GO TO 65
      C
      C   TO FACILITATE THE PROCEDURE OUTLINED IN SUBROUTINE TGTZON,
      C   THE X-Y COORDINATE SYSTEM IS TRANSLATED SO THAT ALL TARGETS
      C   ARE IN THE FIRST QUADRANT, WHEN NECESSARY.
      C
              XM = (XMIN - 999)/1000
              YM = (YMIN - 999)/1000
              XM = - 1000*XM
              YM = - 1000*YM
              XMAX = 32000 - XM
              YMAX = 32000 - YM
        56    IF ((XM+YM) .GT. 0)  WRITE(6,123) XM, YM
            MTT = 0
            IF (NT .EQ. 0)  GO TO 63
            DO 59 I = 1,NT
            IF (TGT(I,10) .GT. MTT)     MTT = TGT(I,10)
        59  CONTINUE
            DO 62  M = 1, MTT
            FLAG = 0
              DO 62  I = 1,NT
            IF (TGT(I,10) .NE. M)  GO TO 62
            IF (FLAG .EQ. 1)  GO TO 60
            IF (NPRINT .LE. 0)  WRITE(6,132) M
            FLAG = 1
        60    IF ((DIM(I,1) .LE. XMAX).AND.(DIM(I,2) .LE. YMAX))  GO TO 61
      C
      C   IF A TARGET IS OUTSIDE THE ALLOWED 32000 X 32000 AREA, IT
      C   IS SHIFTED TO THE EDGE OF THAT AREA AND THE USER IS NOTIFIED.
      C
              IF (DIM(I,1) .GT. XMAX)  DIM(I,1) = XMAX
              IF (DIM(I,2) .GT. YMAX)  DIM(I,2) = YMAX
              WRITE(6,124) I
        61    TGT(I,1) = DIM(I,1) + XM
              TGT(I,2) = DIM(I,2) + YM
```

```
          IF (NPRINT.LE.0)    WRITE (6,112) I,(TGT(I,J),J=1,2),
         X          (TGT(I,J),J=12,13),(TGT(I,J),J=9,11),TGT(I,14),
         X          (NAME(I,L),L=1,NAMES)
   62       CONTINUE
          IF (LIST1 .EQ. 1)    GO TO 65
   63     IF (NT2 .EQ. 0)  GO TO 65
          WRITE(6,115)
          DO 64 I = 1,NT2
          TGT2(I,1) = DIM2(I,1) + XM
          TGT2(I,2) = DIM2(I,2) + YM
   64     WRITE (6,113) I,(TGT2(I,J),J=1,2),(TGT2(I,J),J=12,13),
         X   (TGT2(I,J),J=9,11)
C
   65     NEWTGT = 0
          IF (LIST2 .EQ. 1)    GO TO 95
          WRITE (6,104)
          KPTI = 0
          DO 68 I = NAO, NA
          ATT(I,2) = ATT(I,2) + XM
   68     ATT(I,3) = ATT(I,3) + YM
          DO 70  I = 1, NA
          IF (EMD(ATT(I,9),1,1) .GE. 0)    KPTI = 1
   70     WRITE (6,103) I, (ATT(I,J),J=1,6),ATT(I,10),(ATT(I,J),J=7,9)
         X  , ATT(I,11)
          IF ((ND .EQ. 0).OR.(NPRINT .GT. 0))    GO TO 95
          WRITE (6,105) (I, I=1,10)
          DO 90   I = 1, MWPN
          NR = MTYPE(I)
          IF (NR .EQ. 0)  GO TO 90
          WRITE(6,106) I, WPNREL(I),(EMD(I,J,1),J=1,10)
          WRITE (6,107)   ((EMD(I,J,K), J = 1,10), K=2,NR)
   90     CONTINUE
          IF (CARDS(I) .EQ. 1)  GO TO 90
          CBU = 0
          IF (EMD(I,1,1) .LT. 0)  CBU = 1
          NF = CBU + 1
          WRITE(6,127)
          WRITE(6,107)   ((EMD(I,J,K), J=11,20),K=NF,NR)
   90     CONTINUE
   95     CONTINUE
C
C        TARGET TYPE #1 IS RESERVED FOR RUNWAYS AND TAXIWAYS (OR OTHER
C        LARGE TARGETS IF MOR=0) AND HIT STORAGE IS PROVIDED FOR 250 HITS
C        BUT FOR A MAXIMUM OF FIVE TARGETS OF TYPE #1.
C
          NTX = 0
          DO 99   I = 1, NT
          IF (TGT(I,10) .NE. 1.)    GO TO 99
          NTX = NTX + 1
          IF (NTX .GT. 5)    GO TO 150
          NRW(NTX) = I
   99     CONTINUE
          IF (KTEST .LT. 8)  RETURN
          WRITE(6,112) NAMES
          WRITE(6,118)((STOCKS(I,J),I=1,30),J=1,3)
          WRITE(6,118)((PEOPLE(I,J),I=1,30),J=1,4,3)
          WRITE(6,118)((PEOPLE(I,J),I=102,131),J=1,4,3)
          WRITE(6,118)  (AGE(I,1),I=1,30)
          WRITE(6,118)  (PARTS(I,1),I=1,30)
          WRITE(6,118)  (AMMO(I,1),I=1,26)
          WRITE(6,118)  (TRAP(I,1),I=1,26)
          WRITE(6,118)  (MATERL(I,1),I=1,26)
          WRITE(6,118)  (POL(I,1),I=1,10)
          WRITE(6,118)  (EQUIV(I),I=1,60)
            IF (KTEST .GT.15)    STOP
          RETURN
C
  120      WRITE (6,108)
           STOP
  130      WRITE (6,109)
           STOP
  140      WRITE (6,110)
           STOP
  150      WRITE (6,116)
           STOP
  160      WRITE(6,117)
           STOP
```

```
 170   WRITE(6,119)
       STOP
C
 100  FORMAT(' ',40X,'*****  BASE COMPLEX NAME - ',2A4,'  *****',/,/)
 101  FORMAT( A4,I2,11I6,4A4)
 102  FORMAT (' ', 20X,'TARGET DATA',/,'      NUMBER       X-DIM      Y-DI
     XM    NE LIMB   SE LIMB     ANGLE  TGT TYPE      STORE      BLDG NO',
     X     // )
 103  FORMAT(' ',I4,2X,11I10 )
 104  FORMAT(' ',/,  20X,'ATTACK DATA',/,' NUMBER        HDG     X-DMPI
     X  Y-DMPI        REP        DEP     R-DISP     D-DISP    NO WPNS     LENG
     XTH  WPN TYPE   ARRIVAL',/,/)
 105   FORMAT('1', 35X,'MISS DISTANCE AND KILL PROBABILITY DATA',/,
     X50X,'TARGET TYPES',/,20X,10(6X,I2,2X),/,' WPN TYPE   WPN REL ',
     X/,/,/)
 106  FORMAT('0',I6,7X,F5.3,10I10)
 107  FORMAT(' ', 18X,10I10)
 108  FORMAT(' ',/,'*****  TOO MANY TARGETS HAVE BEEN SPECIFIED  ****')
 109  FORMAT(' ',/,'*****  TOO MANY ATTACKS HAVE BEEN SPECIFIED ****')
 110  FORMAT(' ',/,'*****  TOO MANY TYPES OF WEAPONS HAVE BEEN ',
     X  'SPECIFIED  *****')
 111  FORMAT ('1',  130('*'),/,
     X  40X,'***  TSARINA  ***  DAMAGE ASSESSMENTS FOR TSAR',//,
     X  37X,'BASED ON THE AIDA MODEL OF AIRBASE DAMAGE ASSESSMENT',/,
     X  38X,'DEVELOPED BY D. E. EMERSON AT THE RAND CORPORATION',/
     X  ,130('*'),/,' NO OF TRIALS ', I3,' NPRINT ',I2,'  DAMAGE ',I2,
     X  '  MODE ',I2,
     X  '  MCL ',I6,'(',I4,')   MCW ',I4,'(', I2,')   MIN REPAIR ',
     X I1,'  PLOT HITS ',I1,'   TEST ',I2,/,130('*'),//,55X,'CASE #',
     X  I3,//,20X,'IN THE TSAR SIMULATION, THIS ATTACK AND DAMAGE ',
     X  'OCCUR AT BASE #',I2,' ON DAY',I3,' AT ',I2,':',I2,//)
 112  FORMAT (' ', I4,4X,8I10,2X,4A4)
 113  FORMAT (' ', I4,4X,7I10)
 114  FORMAT ( 12X, 10I6 )
 115  FORMAT('0',20X,' ****   SHELTERS   **** ' ,/ )
 116  FORMAT (' ',/,'*****  TOO MANY RUNWAYS/TAXIWAYS HAVE BEEN',
     X  ' SPECIFIED  ****')
 117  FORMAT(' ',/,'*****  THERE ARE TOO MANY FACILITY CONTENTS ',
     X     'DESCRIPTORS *****')
 118  FORMAT(' ',30I4,/,' ',30I4)
 119  FORMAT('0','TARGETS MAY NOT BE CHANGED FROM CASE TO CASE' )
 121  FORMAT('0',10X,'RESOURCE STORAGE DATA',/,' ','TARGET NUMBER',
     X        5X,'NAME' )
 122  FORMAT(' ', 6X, I4, 8X, 4A4, 2I7 )
 123  FORMAT(' ','ALL TARGET LOCATION DIMENSIONS WERE INCREASED BY',
     X  I6,' IN THE X-DIMENSION AND ',I6,' IN THE Y-DIMENSION',//)
 124  FORMAT('0',' ONE OR BOTH DIMENSIONS OF TARGET ',I4,' WERE ',
     X  'MODIFIED TO PLACE THE TARGET AT THE EDGE OF THE ALLOWED AREA')
 127  FORMAT('0' )
 128  FORMAT('0','STORAGE OF THE RESOURCE LOCATIONS REQUIRED ',I4,
     X   ' LOCATIONS IN THE STOCKS ARRAY,',/,' ','AND THE EQUIVALENCE',
     X   ' DATA USED ',I4,' ELEMENTS OF THE EQUIV ARRAY.',/ )
 129  FORMAT('0','DATA WERE ENTERED FOR ',I4,' TARGETS AND ',I3,
     X              ' ATTACKS.',//)
 132  FORMAT('0', 25X,'** TARGET TYPE # ',I3,'  **',/)
1101  FORMAT(' ',A4,I4,11I8,4A4)
1102  FORMAT(' ','UNIDENTIFIED CARD IMAGE: ',A4,I2,11I6,4A4 )
      END
```

```
            SUBROUTINE TGTDIM
              IMPLICIT INTEGER *2 (A-Z)
              INTEGER *4  MSTAT
              REAL     *4  THETA, S, C, SIN, COS, STAT
C       THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
//            DD     DSN=*.STEP1.COMMON,DISP=SHR
//            DD      *,DCB=BLKSIZE=800
            COMMON /STATS / STAT(500,17),MSTAT(8),NTRIAL,ITRIAL,NSTAT
            IF (NPRINT .LT. 0)    WRITE (6,104)
            IF (NT .EQ. 0)  GO TO 22
            DO 20 I = 1, NT
C       TGTDIM COMPUTES AND STORES THE LOCATION OF THE OTHER CORNERS.
            L1 = TGT(I,12)
            L2 = TGT(I,13)
            THETA = TGT (I, 9)/57.3
            S = SIN (THETA)
            C = COS (THETA)
            L1S = L1*S
            L1C = L1*C
            L2S = L2*S
            L2C = L2*C
            TGT (I,3) = TGT (I, 1) + L1S
            TGT (I,4) = TGT (I, 2) + L1C
            TGT (I,5) = TGT (I, 3) + L2C
            TGT (I,6) = TGT (I, 4) - L2S
            TGT (I,7) = TGT (I, 5) - L1S
            TGT (I,8) = TGT (I, 6) - L1C
            IF ((KTEST .GT. 2) .OR. (NPRINT .LT. 0))
         X    WRITE (6,102) I,(TGT(I,K), K=1,8)
     20     CONTINUE
     22     IF (NT2 .EQ. 0)  GO TO 28
            DO 26 I = 1, NT2
C       TGTDIM COMPUTES AND STORES THE LOCATION OF THE OTHER CORNERS.
            L1 = TGT2(I,12)
            L2 = TGT2(I,13)
            THETA = TGT2(I, 9)/57.3
            S = SIN (THETA)
            C = COS (THETA)
            L1S = L1*S
            L1C = L1*C
            L2S = L2*S
            L2C = L2*C
            TGT2(I,3) = TGT2(I, 1) + L1S
            TGT2(I,4) = TGT2(I, 2) + L1C
            TGT2(I,5) = TGT2(I, 3) + L2C
            TGT2(I,6) = TGT2(I, 4) - L2S
            TGT2(I,7) = TGT2(I, 5) - L1S
            TGT2(I,8) = TGT2(I, 6) - L1C
            IF ((KTEST .GT. 2) .OR. (NPRINT .LT. 0))
         X    WRITE (6,102) I,(TGT2(I,K), K=1,8)
     26     CONTINUE
     28     IF (NT .EQ. 0)  GO TO 50
            NR = 0
            DO 30 I = 1, NT
C       FOR SPECIFIED TARGETS THE TARGET NUMBER IS STORED IN MHIT FOR
C       LATER REFERENCE.
            IF ((TGT(I,11) .LT. 1) .OR. (TGT(I,10) .EQ. 1))    GO TO 30
            NR = NR + 1
            IF (NR .GT. NST)  GO TO 80
            MHIT(NR) = I
            IF ((KTEST .GT. 4) .AND. (ITRIAL .LT. 2))
         X       WRITE (6,101)  NR, MHIT(NR)
     30     CONTINUE
            IF (NR .EQ. NST)    GO TO 50
            NR1 = NR+1
            DO 40   I = NR1, NST
     40     MHIT(I) = 0
     50     CONTINUE
            IF ((NPRINT .LT. 0) .OR. (KTEST .GT. 2))  WRITE(6,104)
            RETURN
     80     WRITE (6,103)
            STOP
    101     FORMAT( ' ','MHIT(' , I2, ') = ',I2)
    102     FORMAT(' ','TARGET CORNER : TGT # ',I4,4(4X,I6,1X,I6))
    103     FORMAT('0', '  COMPUTATION STOPPED:  HIT DATA SPACE ',
         X  'REQUIRED FOR MORE THAN "NST" TARGETS')
    104     FORMAT ('1')
              END
```

```
      SUBROUTINE TGTORD
        IMPLICIT INTEGER *2 (A-Z)
C     THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
//         DD     DSN=*.STEP1.COMMON,DISP=SHR
//         DD     *,DCB=BLKSIZE=800
C     THIS ROUTINE CREATES AN ARRAY IN WHICH THE TARGET NUMBERS ARE
C     ORDERED ACCORDING TO INCREASING VALUES OF (X+Y) OF THE INDEX CORNER
C     OUTPUT VALUES : TO(I,1) IS THE (X+Y) AND TO(I,2) IS THE 'ORIGINAL'
C     TARGET NUMBER;  I.E.  ITS POSITION IN THE INPUT LIST
      DO 10  I=1,NT
C     INITIALIZES TO  (TARGET ORDER) ARRAY
      IF (TGT (I,9) .GT. 45)  GO TO 5
      TO(I,1)=TGT(I,1)+TGT(I,2)
      GO TO 8
    5 TO(I,1)=TGT(I,7)+TGT(I,8)
    8 TO(I,2)=I
   10 CONTINUE
        IF (NT .EQ. 1)  GO TO 25
      DO 20 J=2,NT
      NTEST=0
      DO 15 K=2,NT
C     REORGANIZES THE TO ARRAY INTO INCREASING VALUES OF THE INDEX CORNER
      I=NT-K+2
      IF (TO(I,1)  .GE. TO(I-1,1))  GO TO 15
      NTEST=1
      T=TO(I-1,1)
      TN=TO(I-1,2)
      TO(I-1,1)=TO(I,1)
      TO(I-1,2)=TO(I,2)
      TO(I,1)=T
      TO(I,2)=TN
   15 CONTINUE
      IF (NTEST .EQ. 0)   GO TO 25
   20 CONTINUE
   25 CONTINUE
      IF (KTEST .LT. 3)   GO TO 40
      DO 30   I = 1, NT
      NTO = TO(I,2)
   30 IF (KTEST .GT. 5)   WRITE (6,101)   I , NTO
   40 CONTINUE
      NT1 = NT + 1
      DO 50   I = NT1, NTM
      DO 50   J = 1,2
   50 TO(I,J) = 0.0
      RETURN
  101 FORMAT (' ','  RANK  ',I3,'   TARGET # ', I3)
      END
```

```
          SUBROUTINE TGTZON
            IMPLICIT INTEGER *2 (A-Z)
C       THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
//          DD     DSN=*.STEP1.COMMON,DISP=SHR
//          DD     *,DCB=BLKSIZE=800
C
C       TGTZON IDENTIFIES TARGET LOCATION IN TERMS OF ITS IZONE SO THAT THE
C       SUBSEQUENT SEARCH PROCESS CAN BE REDUCED.   CONSIDER THE ENTIRE
C       TARGET AREA MAPPED BY LINES OF CONSTANT (X+Y).  ALL TARGETS WITH '
C       'INDEX CORNER'  (X+Y)  FALLING INTO THE K-TH 500 FOOT SEGMENT OF
C       (X+Y) ARE IN THE K-TH ZONE.   THE ORDERED INDEX NUMBER FOR THE
C       TARGET WITH THE LOWEST (X+Y) IN THE ZONE IS IZONE(K,1); THAT WITH
C       THE HIGHEST, IS IZONE(K,2).  IF THERE ARE NO TARGETS IN A ZONE, THE
C       IZONE VALUES ARE BOTH EQUAL TO THE INDEX NUMBER OF THE LAST TARGET
C       AS IZONE(K-1,2)).
C
C       TO FACILITATE THIS PROCEDURE THE TARGET COORDINATE SYSTEM IS
C       TRANSLATED, WHEN NECESSARY, SO THAT ALL TARGETS ARE IN THE
C       FIRST QUADRANT.  IF ANY TARGET COORDINATE EXCEEDS 32000, THE
C       TARGET IS PLACED AT THE EDGE OF THE ALLOWED 32000 X 32000 AREA,
C       AND THE USER IS NOTIFIED.
C
          DO 10 I=1,NT
     10   IZ(I) = TO(I,1)/500
          IZONE(1,1) = 0
          IZONE(1,2) = 0
          IF (IZ(1) .NE. 0)    GO TO 14
          K = 1
          IZONE(1,1) = 1
          IZONE(1,2) = 1
            IF (NT .EQ. 1)  GO TO 31
          DO 12  I = 2,NT
          IF (IZ(I) .GT. 0)    GO TO 22
     12   IZONE(1,2) = I
C       ALL HITS IN ZONE #1 HERE.
          GO TO 30
     14   DO 16   K = 2,100
          IF (IZ(1) .EQ. (K-1))    GO TO 18
          IZONE(K,1) = 0
     16   IZONE(K,2) = 0
     18   CONTINUE
          IZONE(K,1) = 1
          IZONE(K-1,2) = 1
          IZONE(K,2) = 1
C       AT THIS POINT K IS ZONE OF FIRST HIT
            IF (NT .EQ. 1)  GO TO 31
          DO 20   I = 2,NT
          IF (IZ(I) .GT. (K-1))    GO TO 22
     20   IZONE(K,2) = I
C       ON TRANSFER TO '22' K IS FIRST OCCUPIED ZONE AND I IS FIRST HIT
C       IN (K+1) ZONE.
     22   CONTINUE
          N = I
          DO 28   I = N,NT
C       SKIP TO 26 IF HIT IN ZONE OF PRIOR HIT
          IF (IZ(I) .EQ. IZ(I-1))    GO TO 26
     24   K = K+1
          IZONE(K,1) = I-1
          IZONE(K,2) = I-1
C       IF NO HITS IN ZONE INCREMENT ZONE
          IF (IZ(I) .GT. (K-1))    GO TO 24
          IZONE(K,1) = I
          IZONE(K,2) = I
          GO TO 28
C       INCREMENT UPPER HIT IN ZONE
     26   IZONE(K,2) = I
     28   CONTINUE
     30   CONTINUE
          IF ((K+1) .GT. 100)    GO TO 36
C       FILL ALL EXCESS ZONES
     31   K1 = K + 1
          DO 32   L = K1,100
          IZONE(L,1) = NT + 1
     32   IZONE(L,2) = NT + 1
     36   CONTINUE
          IF (KTEST .LT. 3)    GO TO 50
          WRITE (6,101)
```

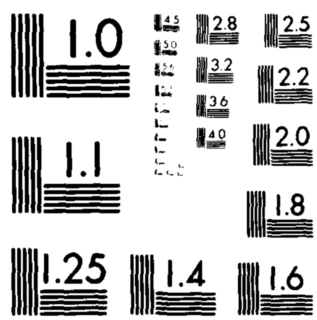
```
          DO 40   K = 1, 100
      40  WRITE (6,102)  K, IZONE(K,1),IZONE(K,2),TO(IZONE(K,1),2),
         X TO(IZONE(K,2),2)
      50  CONTINUE
          RETURN
     101  FORMAT ('1',   'TARGETS BY ZONE',/,
         X'ZONE   LOWER  UPPER      (LOWER  UPPER)')
     102  FORMAT (' ',  I4,4X,I3,4X,I3,7X,I4,3X,I4)
          END
```

```
            SUBROUTINE BOMB
              IMPLICII INTEGER *2 (A-Z)
              REAL*4  RAN,PHI,S,C,SIGR,SIGD,SIGX,SIGY,BDGZX,BDGZY,D,DX,DY,
          X           RERR, DERR, X, Y, SIN, COS
C      THE FOLLOWING JCL INSEPTS TSARINA'S COMMON "BASIC"
//            DD      DSN=*.STEP1.COMMON,DISP=SHR
//            DD      *,DCB=BLKSIZE=800
            COMMON /CBUHIT/ CBUHT(200,2), IR(50), KCBU, KFTI
            IF (KTEST .GT. 2)    WRITE (6,102)
            NCBUHT = 0
            KCBU   = 0
            DO 40 I=1,NA
            IR(I) = 0
            NW = ATT(I,9)
            EMDW = EMD(NW,1,1)
            IF ((MODE .LT. 0) .OR. (ATT(I,11) .EQ. 100))    GO TO 3
C            IF  (MODE .EQ. 0)    GO TO 1
C            RN = 1 + 100*RANDT(1.)
C            GO TO 2
C   1        RN = 1 + 100*RAND(1.)
             RN = 1 + 100*RAN(1)
    2      IF (RN .LE. ATT(I,11))    GO TO 3
            IR(I) = 1
            GO TO 40
    3       CONTINUE
            IF (EMDW .LT. 0.0)     KCBU = 1
            DX=0
            DY=0
            NS=ATT(I,7)
            PHI=ATT(I,1)/57.[illegible]
            S=SIN(PHI)
            C=COS(PHI)
            SIGR=1.48[illegible]*ATT(I,4)
            SIGD=1.48[illegible]*ATT(I,5)
            CALL GAUSS(SIGR,RERR,KTEST,MODE)
            CALL GAUSS(SIGD,DERR,KTEST,MODE)
            AGZX=ATT(I,2)+RERR*S+DERR*C
            AGZY=ATT(I,3)+RERR*C-DERR*S
            BDGZX=AGZX-S*ATT(I,8)/2.
            BDGZY=AGZY-C*ATT(I,8)/2.
            IF (NS .LT. 2)   GO TO 10
            D=ATT(I,8)/(NS-1)
            DX=S*D
            DY=C*D
   10       CONTINUE
           SIGX = ATT(I,6)
           SIGY = ATT(I,10)
            DO 20 M=1,NS
            IF (EMDW .LT. 0)    NCBUHT = NCBUHT + 1
            IF (NCBUHT .GT. 200)    GO TO 60
           IF (MODE .LT. 0)    GO TO 13
C           IF (MODE .EQ. 0)    GO TO 11
C           RN = RANDT(1.)
C           GO TO 12
C    11     RN = RAND(1.)
            RN = RAN(1)
C    12     IF (RN .GT. WPNPPL(NW))    GO TO 17
            IF (RN .GT. WPNREL(NW))    GO TO 17
    13     CONTINUE
           CALL GAUSS(SIGX,X,KTEST,MODF)
           CALL GAUSS(SIGY,Y,KTESI,MODE)
            BAGZX=BDGZX+X*S+Y*C
            BAGZY=BDGZY+X*C-Y*S
            IF ((KTEST .GT. 0) .OR. (NPRINI .LT. -1))
         X        WRITE (6,101) I, M, BAGZX, BAGZY
            IF (EMDW .GE. 0)    GO TO 16
            CBUHT(NCBUHT,1) = BAGZX
            CBUHT(NCBUHT,2) = BAGZY
            GO TO 18
   16       IF ((BAGZX+BAGZY) .LT. -500)    GO TO 18
            INDEX = I
            CALL TESTHT(INDEX,BAGZX,BAGZY,NW)
            GO TO 18
   17       IF (EMDW .LT. 0)    CBUHT(NCBUHT,1) = -10000
   18       BDGZX=BDGZX+DX
            BDGZY=BDGZY+DY
```

AD-A090 682 RAND CORP SANTA MONICA CA F/6 1/5
TSARINA: USER'S GUIDE TO A COMPUTER MODEL FOR DAMAGE ASSESSMENT--ETC(U)
JUL 80 D EMERSON F49620-77-C-0023
UNCLASSIFIED RAND/N-1460-AF NL

2 of 2
AD A
090682

END
DATE
FILMED
11-80
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU

```
   20   CONTINUE
   40   CONTINUE
        RETURN
   60   WRITE(6,103)
        STOP
  101   FORMAT(' ','ATTACK #',I4,'   BOMB #',I4,'   X-DIM ',I8,
     X    '  Y-DIM  ', I8 )
  102   FORMAT ('1',30('*'),'  BOMB IMPACT AND HIT DATA  ',30('*'),/,/)
  103   FORMAT('0','THE CBUHT ARRAY MUST BE ENLARGED TO ',
     X    'ACCOMMODATE MORE CBU WEAPONS')
        END
```

```
            SUBROUTINE  TESTHT(I,BX,BY,NW)
              IMPLICIT INTEGER *2 (A-Z)
              INTEGER *4 AID, HELP, MINO, MAXO, XY, K
              REAL     *4   DMAX, DR, PP
      C     THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
      //      DD     DSN=*.STEP1.COMMON,DISP=SHR
      //      DD     *,DCB=BLKSIZE=800
      C
            DIMENSION  FLAG(6), DR(9), PP(6)
      C
            XY = BX + BY
            NN=XY/500
            K = MAXO(NN+1,1)
            K = MINO(K,99)
            LL = 0
            IF (K .EQ. 1)  GO TO 10
            LL = IZONE (K-1, 1)
            IF (K .EQ.2)  GO TO 15
            IF ((IZONE(K-1,1) .EQ. IZONE(K-2,2)) .AND.
           X  (IZONE(K-2,2) .NE. 1))     LL = IZONE(K,1)
            GO TO 15
         10 LL = IZONE (1, 1)
         15 CONTINUE
            LU = IZONE ((K+1), 2)
            IF (KTEST .GT. 3)   WRITE (6,102)  I, LL,LU
            DO 100 IL = LL, LU
      C     CONSIDER ALL TARGETS BETWEEN THE LIMITS OF LL AND LU
            L = TO(IL,2)
            IF (L .LE. 0)   GO TO 100
            IF ((TGT(L,12) + TGT(L,13)) .GT. 500)   GO TO 100
            NTGT = TGT(L,10)
            MFLAG = 0
            IF ((NTGT .NE. SHELT).OR.(ALLMC .NE. 1))  GO TO 20
            MFLAG = 1
            DR(9) = EMD(NW,NTGT,10)
         20 DR(1) = EMD(NW,NTGT,1)
            DMAX = DR(1)
            DR(2)  = EMD(NW,NTGT,2)
            IF (DMAX .LT. DR(2))  DMAX = DR(2)
            HELP = EMD(NW,NTGT,9)
            DO 25  FLG = 1,5
            AID = 10**(6-FLG)
            FLAG(FLG) = HELP/AID
         25 HELP = HELP - AID*FLAG(FLG)
            FLAG(6) = HELP
      C
            DO 30   FLG = 1,6
            DR(FLG+2) = 0
            IF (FLAG(FLG) .LT. 4)  GO TO 30
              IF (FLAG(FLG) .LT. 6)  GO TO 28
              AID = EMD(NW,NTGT,(FLG+2))
              DR(FLG+2) = AID/1000
              PP(FLG) = (AID - 1000*DR(FLG+2))/100.
              IF (FLAG(FLG) .EQ. 8)  GO TO 30
              GO TO 29
         28 DR(FLG+2) = EMD(NW,NTGT,(FLG+2))
            IF (FLAG(FLG) .NE. 4)  GO TO 30
         29 IF (DR(FLG+2) .GT. DMAX)  DMAX = DR(FLG+2)
         30 CONTINUE
         40 CONTINUE
              D  = DMAX
              D  = 1.414*D
            IF ((TGT(L,1) - D) .GT. BX)    GO TO 100
            IF ((TGT(L,4) + D) .LT. BY)    GO TO 100
            IF ((TGT(L,5) + D) .LT. BX)    GO TO 100
            IF ((TGT(L,8) - D) .GT. BY)    GO TO 100
            IF (KTEST .GT. 4) WRITE (6,101) I,L,BX,BY,DMAX,HELP
            CALL HITTGT(I,L,BX,BY,NW,NTGT,DR,PP,FLAG,MFLAG)
        100 CONTINUE
            MFLAG = 0
            DO 120  L = 1,NT
            IF ((TGT(L,12) + TGT(L,13)) .LE. 500)   GO TO 120
            NTGT = TGT(L,10)
            DR(1) = EMD(NW,NTGT,1)
            DMAX = DR(1)
            IF ((NTGT .EQ. 1).AND.(MCR .GT. 0))  GO TO 115
```

```
      DR(2) = EHD(NW,NTGT,2)
      IF (DMAX .LT. DR(2))  DMAX = DR(2)
      HELP = EHD(NW,NTGT,9)
      DO 105  FLG = 1,5
      AID = 10**(6-FLG)
      FLAG(FLG) = HELP/AID
  105 HELP = HELP - AID*FLAG(FLG)
      FLAG(6) = HELP
C
      DO 110   FLG = 1,6
      DR(FLG+2) = 0
      IF (FLAG(FLG) .LT. 4)  GO TO 110
        IF (FLAG(FLG) .LT. 6)  GO TO 106
        AID = EHD(NW,NTGT,(FLG+2))
        DR(FLG+2) = AID/1000
        PP(FLG) = (AID - 1000*DR(FLG+2))/100.
        IF (FLAG(FLG) .EQ. 8)  GO TO 110
        GO TO 108
  106 DR(FLG+2) = EHD(NW,NTGT,(FLG+2))
      IF (FLAG(FLG) .NE. 4)  GO TO 110
  108 IF (DR(FLG+2) .GT. DMAX)  DMAX = DR(FLG+2)
  110  CONTINUE
  115 IF (KTEST .GT. 4)    WRITE (6,101) I,L,BX,BY,DMAX,HELP
      CALL HITTGT(I,L,BX,BY,NW,NTGT,DR,PP,FLAG,MFLAG)
  120 CONTINUE
      RETURN
  101 FORMAT (' ',10X,'TESTHIT:  ATTACK ',I3,'      TGT ',I4,  '   X-DIM
     X ', I6 , '   Y-DIM ', I6 , F8.1, I8 )
  102 FORMAT(' ',20X,'ATTACKER ',I3,'  TARGET RANK LIMITS ',2I6,/)
      END
```

```
      SUBROUTINE HITTGT(I,L,BX,BY,NW,NTGT,DP,PP,FLAG,MFLAG)
        IMPLICIT INTEGER *2 (A-Z)
        INTEGER *4  AID
        REAL    *4 DD,DR,S,C,CO,T,D,RHO,F,FX,FY,FAC,F1,F2,PP,
     X  TEST,  OLD1,OLD2,  YD,A,AL,DN,D2,R,RL,PS,TOT,Z, SIN, COS
C    THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
//      DD     DSN=*.STEP1.COMMON,DISP=SHR
//      DD     *,DCB=BLKSIZE=800
      DIMENSION DN(4), F(4), IS(4), D2(4), DR(9), PP(6), FLAG(6),FAC(9)
C
      INSIDE = 0
      HIT1 = 0
      DO 2  NR = 1,9
    2 FAC(NR) = 0.0
      S = SIN (TGT (L,9)/57.3)
      C = COS (TGT (L,9)/57.3)
      CO= C
C
        D = 0
        T = 0
        F1 = 0.0
        F2 = 0.0
        NSTART = 1
        NEND   = 9
C
    4  CONTINUE
C
      DO 100  NP = NSTART, NEND
      IF (NP .GT. 1)  GO TO 5
      DD = 0
      GO TO 30
    5 NR = NP - 1
      IF (NR .GT. 2)  GO TO 10
      IF ((NTGT.EQ.1).AND.(MCR.NE.0).AND.(NR.GT.1))  GO TO 105
      IF ((NR .EQ. 2).AND.(DR(NR) .EQ. 0.0))  GO TO 100
      DD = DR(NR)
      GO TO 20
   10 IF (NR .LT. 9)  GO TO 15
      DD = DR(9)
      GO TO 20
   15 IF (FLAG(NR-2) .LT. 4)  GO TO 85
      DD = DR(NR)
      IF ((FLAG(NR-2) .EQ. 5).AND.(HIT1 .EQ. 0))  GO TO 90
      IF ((FLAG(NP-2) .EQ. 8).AND.(HIT1 .EQ. 0))  GO TO 90
   20 IF (DD .EQ. 0.0) GO TO 30
      T = (S + CO)*DD
      D = (S - CO)*DD
   30 X1 = TGT (L,1) - T
      Y1 = TGT (L,2) + D
      X2 = TGT (L,3) + D
      Y2 = TGT (L,4) + T
      X3 = TGT (L,5) + T
      Y3 = TGT (L,6) - D
      X4 = TGT (L,7) - D
      Y4 = TGT (L,8) - T
     IF ((BX .LT. X1) .OR. (BX .GT. X3))  GO TO 100
     IF ((BY .LT. Y4) .OR. (BY .GT. Y2)) GO TO 100
      IF (TGT (L, 9) .EQ. 0)  GO TO 50
      T = S/CO
      C = 1./T
      IF ((BX .LT. X2) .AND. (BY .GT. (Y1+C*(BX-X1))))   GO TO 100
      IF ((BX .GT. X2) .AND. (BY .GT. (Y2-T*(BX-X2))))   GO TO 100
      IF ((BX .GT. X4) .AND. (BY .LT. (Y4+C*(BX-X4))))   GO TO 100
      IF ((BX .LT. X4) .AND. (BY .LT. (Y1-T*(BX-X1))))   GO TO 100
   50 CONTINUE
      IF (NP .NE. 1)  GO TO 56
      INSIDE = 1
      IF ((EMD(NW,NTGT,10) .EQ. -1).OR.(NTGT .EQ. SHPLT))  GO TO 100
      DR(1) = EMD(NW,NTGT,10)
      DR(2) = EMD(NW,NTGT,11)
      DMAX = DR(1)
      IF (DMAX .LT. DR(2))  DMAX = DR(2)
      DO 54  FLG = 1,6
      DR(FLG+2) = 0
      IF (FLAG(FLG) .LT. 4)  GO TO 54
        IF (FLAG(FLG) .LT. 6)  GO TO 51
        AID = EMD(NW,NTGT,(FLG+11))
```

```
           DR(FLG+2) = AID/1000
           PP(FLG) = (AID - 1000*DR(FLG+2))/100.
           IF (FLAG(FLG) .EQ. 8)  GO TO 54
           GO TO 52
   51    DR(FLG+2) = EMD(NW,NTGT,(FLG+11))
         IF (FLAG(FLG) .NE. 4)  GO TO 54
   52    IF (DR(FLG+2) .GT. DMAX)  DMAX = DR(FLG+2)
   54    CONTINUE
         GO TO 100
   56    IF((KTEST.GT.2).OR.(NPRINT.LT.0))WRITE(6,101)L,BX,BY,INSIDE,NR,DP
         IF (NR .NE. 1)  GO TO 60
         HIT1 = 1
         NHIT(L) = NHIT(L) + 1
         IF (INSIDE .EQ. 0)  OHIT(L) = OHIT(L) + 1
   60    IF ((NTGT .EQ. 1) .AND. (MCR .NE. 0))   GO TO 110
C
C   RESULTS INCLUDE AN ESTIMATE OF THAT FRACTION OF THE TARGET
C   AREA THAT IS COVERED BY CIRCLES OF RADIUS DD1 AND DD2.
C
C   COMPUTE DISTANCES NORMAL TO THE FOUR SIDES OF THE TARGET.
C
         IF (TGT(L,9) .EQ. 0)    GO TO 62
         YD = Y1 + C*(BX-X1) - BY
         DN(1) = S*YD
         YD = Y2 - T*(BX-X2) - BY
         DN(2) = CO*YD
         YD = BY - Y4 - C*(BX-X4)
         DN(3) = S*YD
         YD = BY - Y1 + T*(BX - X1)
         DN(4) = CO*YD
         GO TO 64
   62    CONTINUE
         DN(1) = BX - X1
         DN(2) = Y2 - BY
         DN(3) = X3 - BX
         DN(4) = BY - Y1
   64    CONTINUE
         TOT = 0.0
         DO 66   N = 1,4
         D2(N) = DD-DN(N)
         IF (D2(N) .LE. 0.0)   GO TO 66
         TOT = TOT + D2(N)*D2(N)
   66    CONTINUE
         IF (KTEST.GT.5)WRITE(6,1102) TOT,(DN(N),N=1,4),(D2(N),N=1,4)
         IF (TOT .LE. DD*DD)   GO TO 68
         IF (NR .NE. 1)  GO TO 100
         HIT1 = 0
         NHIT(L) = NHIT(L) - 1
         OHIT(L) = OHIT(L) - 1
         GO TO 100
C
C   SPECIAL TREATMENT IS REQUIRED WHEN SHELTERS
C   ARE HANDLED WITH THE MONTE CARLO MODE.
C
   68    IF (MFLAG .EQ. 0)  GO TO 74
         NS = TGT(L,14) - 1000
         IF (NR .EQ. 1)  GO TO 70
         IF (NR .NE. 9)  GO TO 74
         GO TO 72
   70    P2(NS,3) = 1. - (1. - P2(NS,3))*(100 - EMD(NW,NTGT,12))/100.
         P2(NS,4) = 1. - (1. - P2(NS,4))*(100 - EMD(NW,NTGT,14))/100.
         GO TO 74
   72    P2(NS,5) = 1. - (1. - P2(NS,5))*(100 - EMD(NW,NTGT,13))/100.
         IF (KTEST .GT. 5)  WRITE(6,1101)  NR, (P2(NS,K),K=3,5)
         GO TO 102
   74    CONTINUE
          RL = DD
   80    CONTINUE
         AL = 3.1416*RL*RL
         DO 84   N = 1,4
         F(N) = 0.0
         R = DD - DN(N)
         IF ((R .GT. 0.0) .AND. (R .GE. RL))   F(N) = 1.0
         IF (F(N) .EQ. 1.0)   GO TO 83
         IF ((R .LT. 0.0) .AND. (-R .GT. RL))   GO TO 83
         IF (R .LT. 0.0)   R = -R
         IF (R .EQ. 0.0)   GO TO 82
```

```
          Z = R/RL
          RHO = 2.*ATAN((1./(Z*Z)-1.)**(.5))
          A = RL*RL*(RHO - SIN(RHO))/2.
          IF (DN(N) .GT. DD)   A = AL - A
          F(N) = 1. - A/AL
          GO TO 83
   82     F(N) = .5
   83     IF (KTEST.GT.5) WRITE(6,1101) N,R,Z,A,AL,F(N),DD
   84     CONTINUE
          FX = 1. - F(1) - F(3)
          FY = 1. - F(2) - F(4)
          IF (FX .LT. 0.0)   FX = 0.0
          IF (FY .LT. 0.0)   FY = 0.0
          IF (KTEST .GT.4) WRITE(6,1002)FX,FY,R,RL,A,AL,(F(N),N=1,4)
C       NOTE THAT THE USE OF FX AND FY PROVIDES ONLY AN APPROXIMATE RESULT
          FAC(NR) = FX*FY*AL/(TGT(L,12)*TGT(L,13))
          FAC(NR) = AMIN1(1.,FAC(NR))
            IF (NR .LT. 3) GO TO 844
            IF (FLAG(NR-2) .LT. 6)  GO TO 844
            IF (FLAG(NR-2) - 7)  841, 843, 842
  841       FAC(NR) = (3.0*F1 + FAC(NR))*PP(NR-2)/4.0
            GO TO 844
  842       IF (HIT1 .EQ. 0)  GO TO 844
  843       FAC(NR) = (3.0*F2 + FAC(NR))*PP(NR-2)/4.0
  844       CONTINUE
          PS = 1. - FAC(NR)
          P(L,NR) = 1. - (1. - P(L,NR))*PS
          GO TO 90
   85     XNR = NR
          IF ((EMD(NW,NTGT,10).NE.-1).AND.(NTGT.NE.SHELT)) XNR=NR+9*INSIDE
          IF (FLAG(NR-2) - 2)  86, 88, 87
   86     IF (FLAG(NR-2) .EQ. 0)  GO TO 90
          P(L,NR) = 1. - (1. - P(L,NR))*(1.-(FAC(1)*EMD(NW,NTGT,XNR))/100.)
          GO TO 90
   87     IF (HIT1 .EQ. 0)  GO TO 90
   88     P(L,NR) = 1. - (1. - P(L,NR))*(1.-(FAC(2)*EMD(NW,NTGT,XNR))/100.)
   90     CONTINUE
            IF (NR .EQ. 2)  F2 = FAC(2)
          IF (NR .NE. 1)  GO TO 100
            F1 = FAC(1)
          IF (TGT(L,11) .LT. 1.)   GO TO 100
          IF (NHIT(L) .GT. 25)   GO TO 100
          DO 95   J = 1, NST
          IF (NHIT(J) .EQ. 0)  GO TO 100
          IF (NHIT(J) .NE. L)   GO TO 95
          HIT (J,1,NHIT(L))=BX
          HIT (J,2,NHIT(L))=BY
          HIT (J,3,NHIT(L))=NW
          GO TO 100
   95     CONTINUE
  100     CONTINUE
          IF ((NTGT .NE. SHELT).OR.(NSTART .EQ. 10))  GO TO 102
          NSTART = 10
            NEND = 10
          GO TO 4
  102     CONTINUE
          IF (KTEST .GT. 3) WRITE(6,1001) L,(FAC(K),K=1,8),(P(L,J),J=1,8)
  105     CONTINUE
          RETURN
  110     CONTINUE
          DO 120   J = 1,5
          IF (NRW(J) .EQ. 0)  GO TO 130
          IF (NRW(J) .NE. L)  GO TO 120
          HITR(J,1,NHIT(L)) = BX
          HITR(J,2,NHIT(L)) = BY
          HITR(J,3,NHIT(L)) = NW
  120     CONTINUE
  130     CONTINUE
          RETURN
  101     FORMAT(' ',10X,'***** HITTGT   NHIT(',I3,')',4I6,9F8.3)
 1002     FORMAT(' ', '   FX ',F6.0  ,'   FY ', F6.0, 6X,4F8.0,4F8.4 )
 1001      FORMAT(' ',' TGT #', I3, '  FAC ', 8F6.3, '   PK ',8F6.3)
 1101     FORMAT(' ', I6,9F12.3 )
 1102     FORMAT(' ', F12.3, 8F8.3 )
          END
```

```
              SUBROUTINE GAUSS (S,V,KTEST,MODE)
              INTEGER *2 KTEST,MODE
              IF ((KTEST .GT. 7) .OR. (MODE .LT. 0))    GO TO 50
              A=0.0
C              IF (MODE .EQ. 0)    GO TO 20
              DO 10 I=1,12
              Y = RAN(1)
C              Y = RANDT(1.)
   10         A = A + Y
C              GO TO 40
C   20         DO 30 I=1,12
C              Y=RAND(1.)
C   30         A = A + Y
C   40         V = (A-6.0)*S
              V = (A-6.0)*S
              RETURN
   50         CONTINUE
              V = 0.0
              RETURN
              END
```

```
              SUBROUTINE CHECKR
                IMPLICIT INTEGER *2 (A-Z)
              INTEGER *4  MSTAT, NHOLES
              REAL    *4  STAT
C     THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
//            DD     DSN=*.STEP1.COMMON,DISP=SHR
//            DD     *,DCB=BLKSIZE=800
              COMMON /STATS / STAT(500,17),MSTAT(8),NTRIAL,ITRIAL,NSTAT
              COMMON /CONTRL/ NREP,NPLOT,INW,INL,CHANGE,NSAVE,LIST,NJMEM
              NC=0
              NN=0
              LHOLES = 10000
              DO 40 MRW = 1, 5
C     CYCLE THRU AS MANY AS 8 RUNWAY/TAXIWAYS.
              IRW = NRW(MRW)
C     EXIT IF NO TARGET NUMBER (IRW) FOUND.
              IF (IRW .EQ. 0)  GO TO 50
              NN=NN+1
              IF (NHIT(IRW) .EQ. 0)   GO TO 40
              IF (KTEST .GT. 4)   WRITE (6,102) IRW
               INDEX = MRW
               CALL RUNWAY (INDEX, IRW, ICOND, NHOLES)
C     RUNWAY SUBROUTINE RETURNS ICOND = 0 IF RUNWAY HAS REQUIRED SPACE;
C        ICOND = 1 IF NOT.
C
              IF (ICOND .EQ. 1) NC=NC+1
              IF (NHOLES .LT. LHOLES)    LHOLES = NHOLES
              IF ((NPRINT .EQ. 4) .OR. (NPRINT .EQ. 6))
     X          WRITE(6,103) ITRIAL, IRW, NHIT(IRW),NHOLES
   40         CONTINUE
   50         IF (NC .EQ.NN)    GO TO 60
              IF (NPRINT .LT. 3)    WRITE (6,101)
              NSTAT = NSTAT + 1
   60         CONTINUE
              IF ((NREP .EQ. 1) .AND. (NC .EQ. NN))    GO TO 70
              RETURN
   70         MSTAT(1) = MSTAT(1) + LHOLES
              MSTAT(2) = MSTAT(2) + LHOLES*LHOLES
              REPAIR = LHOLES
              IF (LHOLES .EQ. 10000)  REPAIR = 0
              RETURN
  101         FORMAT(' ',' AT LEAST ONE RUNWAY IS AVAILABLE')
  102         FORMAT(' ',' CHECK TARGET #', I4)
  103         FORMAT(' ','TRIAL',I4,'   TGT',I4,'   HITS',I4,'   REPAIRS',I3)
              END
```

```
      SUBROUTINE RUNWAY (MRW, IRW, ICOND, NHOLES)
        IMPLICIT INTEGER *2 (A-Z)
        INTEGER *4 NHOLES, IHOLE, MSTAT
        REAL    *4  STAT,  TH, TH1, TH2, XX, YY, SIN, COS
C    THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
//      DD    DSN=*.STEP1.COMMON,DISP=SHR
//      DD    *,DCB=BLKSIZE=800
      COMMON /STATS / STAT(500,17),MSTAT(8),NTRIAL,ITRIAL,NSTAT
      COMMON /HITS/ XN(250),YN(250),NZ(250)
      COMMON /CONTRL/ NREP,NPLOT,INW,INL,CHANGE,NSAVE,LIST,NJMEM
      DIMENSION NTEST(250), YH(250,2)
C
C    CHECKS FOR THE EXISTENCE OF THE SPECIFIED RUNWAY MINIMUMS (MCL Y
C    MCW) ON EACH RUNWAY AND DESIGNATED TAXIWAY (TYPE #1 TARGETS).
C    STOPS SEARCHING A GIVEN RUNWAY WHENEVER REQUIREMENT IS SATISFIED.
C
      TH=TGT(IRW,9)/57.3
C
C    ESTABLISH ORIGIN (XO, YO) FOR A RECTANGULAR COORDINATE SYSTEM WITH
C    THE X-AXIS ON THE MORE SOUTHERLY EDGE OF THE RUNWAY.
C
      NHI = 0
      NHOLES =1000
      DO 5   N = 1, 250
    5 NZ(N) = 0
       IF (TGT(IRW,12) .GT. TGT(IRW,13))   GO TO 10
      NDIR=1
      XO=TGT(IRW,1)
      YO=TGT(IRW,2)
      LTH = TGT(IRW,13)
      WID = TGT(IRW,12)
      GO TO 20
   10 NDIR=2
      XO=TGT(IRW,7)
      YO=TGT(IRW,8)
      LTH = TGT(IRW,12)
      WID = TGT(IRW,13)
   20 CONTINUE
     IF (KTEST .GT. 4)  WRITE (6,1004)IRW,XO,YO,LTH,WID,MCL,MCW
      IF (MCW .GT. WID)   GO TO 320
      NHIT1 = NHIT(IRW)
      DO 50   I = 1, NHIT1
      IF (EMD(HITR(MRW,3,I),1,1) .LT. 0)   GO TO 50
      NTW = HITR(MRW,3,I)
      GO TO 60
   50 CONTINUE
   60 CONTINUE
      NON = 1
      EMDW = EMD(NTW,1,1)
      DO 70   I = 1, NHIT1
      IF (HITR(MRW,3,I) .EQ. NTW)   GO TO 70
      IF (EMD(HITR(MRW,3,I),1,1) .LT. 0)   GO TO 70
      NON = 0
      EMDW = 0.0
      GO TO 80
   70 CONTINUE
   80 CONTINUE
      DO 140   I = 1, NHIT1
C    TRANSFORM HIT COORDINATES TO RUNWAY COORDINATES.
      NHI = NHI + 1
      XB = HITR(MRW,1,I)
      YB = HITR(MRW,2,I)
      IF (EMD(HITR(MRW,3,I),1,1) .LT. 0.)   NZ(I) = 1
      IF (TH .EQ. 0.0)  GO TO 110
      XX=XB-XO
      YY=YB-YO
      R = (XX*XX+YY*YY)**(0.5)
      IF (KTEST .GT. 7)  WRITE (6,1010) R,XX,YY
     TH1 = ATAN(YY/XX)
     IF (XX .LT. 0.0)   TH1 = TH1 + 3.1416
     TH2 = TH1 + TH
     IF (NDIR .EQ. 2)   TH2 = TH2 - 1.5706
      XN(I)=R*COS(TH2)
      YN(I)=R*SIN(TH2)
      IF (KTEST .GT. 6)   WRITE(6,1009)I,XN(I),YN(I)
      GO TO 130
  110 IF (NDIR .EQ. 2)  GO TO 120
```

```
          XN(I)=XB-XO
          YN(I)=YB-YO
          GO TO 130
  120     XN(I)=YB-YO
          YN(I)=XO-XB
  130     IF ( I .GT. 249)   GO TO 150
  140     CONTINUE
          GO TO 160
  150     WRITE (6,1001) IRW, ITRIAL
  160     CONTINUE
          NH = NHI
C     IF NPLOT .EQ. 2 RUNWAY IMPACTS ARE PLOTTED FOR ALL CONDITIONS.
C     IF NPLOT .EQ. 1 IMPACTS ONLY PLOTTED WHEN RUNWAY IS CLOSED.
         IF ((MRW .EQ. 1) .AND. (NPLOT .EQ. 2))  WRITE(6,1012)ITRIAL
          IF (NPLOT .EQ. 2)    CALL PLOTHT(NH,IRW,LTH,WID)
          IF (KTEST .GT.6)    WRITE (6,1006) NHIT(IRW),NH
          DO 170 I=1,NH
          YH(I,1) = YN(I)
  170     YH(I,2)=I
          IF (NH .EQ. 1)  GO TO 190
          DO 180 J=2,NH
          DO 180 K=2,NH
C     ORDER ALL HITS FROM LOWEST Y TO HIGHEST.  YH(I,1) IS THE
C     Y COORDINATE, YH(I,2) THE HIT NUMBER, OF THE I TH ORDERED HIT.
          I=NH-K+2
          IF (YH(I,1) .GE. YH(I-1,1))  GO TO 180
          T=YH(I-1,1)
          TN=YH(I-1,2)
          YH(I-1,1)=YH(I,1)
          YH(I-1,2)=YH(I,2)
          YH(I,1)=T
          YH(I,2)=TN
  180     CONTINUE
  190     CONTINUE
          XL = 0
          XU = MCL
  200     YL = 0
          YU = MCW
          NYL = 1
          IF (NON .EQ. 0)    GO TO 210
          YL = YL - EMDW
          YU = YU + EMDW
  210     CONTINUE
  220     IHOLE = 0
          DO 250   I = NYL , NH
          IF (NZ(I) .EQ. 1)    GO TO 250
          YT = YH(I,1)
          IF (NON .EQ. 1)   GO TO 230
          R = EMD(HITR(MRW,3,YH(I,2)),1,1)
          YL = YL - R
          YU = YU + R
  230     IF (YT .LT. YL)   GO TO 240
          XT = XN(YH(I,2))
          IF ((XT .LT. XL) .OR. (XT .GT. XU))    GO TO 240
          IF (YT .GT. YU)   GO TO 260
          IHOLE = IHOLE + 1
          IF (KTEST .GT. 6)    WRITE(6,1008)  I, IHOLE, YL,YU
          NTEST(IHOLE) = I
          IF ((NREP .EQ. 0) .OR. (IHOLE .GT. NHOLES))    GO TO 260
  240     IF (NON  .EQ. 1)    GO TO 250
          IF (I .EQ. NH)    GO TO 260
          YL = YL + R
          YU = YU - R
  250     CONTINUE
  260     CONTINUE
          IF (IHOLE .EQ. 0)    GO TO 300
          IF (NON .EQ. 1)    NYL = NTEST(1)
          NHOLES = MINO(NHOLES, IHOLE)
          IF (NON .EQ. 0)   GO TO 270
          YL = YL + INW
          YU = YU + INW
          IF (YU .GT. (WID+EMDW))   GO TO 280
          GO TO 220
  270     YL = YL + R + INW
          YU = YU - R + INW
          IF (YU .GT. WID)   GO TO 280
           IF (KTEST .GT. 4)   WRITE(6,1008) NHOLES
```

```
      GO TO 220
280   CONTINUE
      XL = XL + INL
      XU = XU + INL
      IF (XU .GT. LTH)   GO TO 290
      GO TO 200
290    CONTINUE
      IF (NPRINT .GE. 3)   GO TO 295
      IF ((MRW .EQ. 1) .AND. (NPLOT .NE. 2))  WRITE (6,1012) ITRIAL
       WRITE(6,1002) IRW
      IF (NREP .GT. 0)   WRITE (6,1011) NHOLES
      IF (NPLOT .EQ. 1)   CALL PLOTHT(NH,IRW,LTH,WID)
295   ICOND = 1
      IF ((NPLOT .GT. 0) .AND. (NPRINT .LT. 3)) WRITE(6,1000)
       RETURN
300   IF (NPRINT .GT. 2)   GO TO 310
      IF (MRW .EQ. 1)   WRITE (6,1012) ITRIAL
      WRITE (6,1003) IRW
310   ICOND = 0
      IF ((NPLOT .EQ. 2) .AND. (NPRINT .LT. 3)) WRITE(6,1000)
      NHOLES = 0
      RETURN
320   WRITE(6,1005)  IRW
      STOP
1000   FORMAT('1')
1001   FORMAT('0','ONLY FIRST 250 HITS TESTED FOR TARGET #',I4,
     X  ' IN TRIAL #',I4,/,/)
1002   FORMAT('0',20X,'RUNWAY #',I3,' IS CLOSED',/,/)
1003   FORMAT('0',20X,'RUNWAY #',I3,' IS OPEN',/)
1004   FORMAT(' ','  RUNWAY SPECS ',I4,4X,6I8)
1005   FORMAT ('0','***** TARGET #',I3,' IS TOO NARROW FOR ',
     X  'FLIGHT OPERATIONS')
1006   FORMAT(' ','    # HITS TO CHECK', 2I6)
1008   FORMAT(' ','  TEST POINT #B ',3I10)
1009   FORMAT(' ','  TEST POINT #E', I4,2I10)
1010   FORMAT(' ','    R = ',F8.0,10X,2F10.0)
1011   FORMAT(' ', 17X,I4,' HOLES MUST BE REPAIRED TO MEET RUNWAY ',
     X  'MINIMUMS',/)
1012   FORMAT ('1', '    *****  TRIAL #',I3,'  *****',/)
      END
```

```
      SUBROUTINE PRINT
        IMPLICIT INTEGER *2 (A-Z)
        INTEGER *4  MSTAT
        REAL    *4  STAT, TOTAL
C    THE FOLLOWING JCL INSERTS ZSARINA'S COMMON "BASIC"
//      DD    DSN=*.STEP1.COMMON,DISP=SHR
//      DD    *,DCB=BLKSIZE=800
      COMMON /STATS / STAT(500,17),MSTAT(8),NTRIAL,ITRIAL,NSTAT
      IF (NTRIAL .EQ. 1)   GO TO 1
      IF ((NPRINT.LT.0).OR.((NPRINT.LT.3).AND.(MCR.EQ.0)) )WRITE(6,107)
      WRITE (6,108)   ITRIAL
      GO TO 2
  1   CONTINUE
      WRITE (6,106)
  2   CONTINUE
      WRITE (6,101)
      DO 10 M=1,MTT
      NN= 0
      FLAG = 0
      DO 10  I=1,NT
      IF (TGT(I,10) .NE. M)  GO TO 10
      NN= NN+1
      NAID = NHIT(I) - NCBU(I)
      TOTAL = NAID + OHIT(I) + COV(I)
      DO 5  J = 1,8
   5  TOTAL = TOTAL + P(I,J)
      IF (TOTAL .EQ. 0.0)  GO TO 10
      IF (FLAG .EQ. 1)  GO TO 8
      FLAG = 1
      WRITE(6,109) M
   8  WRITE(6,102) I,NAID,OHIT(I),COV(I),(P(I,J),J=1,8),TGT(I,14),
     X             (NAME(I,J),J=1,NAMES)
  10  CONTINUE
      IF (NPRINT .GT. -1)   GO TO 60
      WRITE (6,103)
       DO 20   M = 1, NST
      IF (MHIT(M) .EQ. 0)   GO TO 30
      NN=0
      NL = NHIT(MHIT(M))
      IF (NL .EQ. 0)   GO TO 20
      DO 15   N=1, NL
      NN=NN+1
      IF (NN .EQ. 26)   GO TO 20
      IF (NN .EQ. 1) WRITE (6,104)  MHIT(M)
      X=HIT(M,1,NN)
      Y=HIT(M,2,NN)
      NWPN=HIT(M,3,NN)
      WRITE (6,105) X,Y,NWPN
  15  CONTINUE
  20  CONTINUE
  30  CONTINUE
      DO 50   M = 1, 5
      IF (NRW(M) .EQ. 0)   GO TO 60
      NN = 0
      NL = NHIT(NRW(M))
      IF (NL .EQ. 0)   GO TO 50
      DO 40   N = 1, NL
      NN = NN + 1
      IF (NN .EQ. 251)   GO TO 50
      IF (NN .EQ. 1)    WRITE (6,104)   NRW(M)
      X = HITR(M,1,NN)
      Y = HITR(M,2,NN)
      NWPN = HITR(M,3,NN)
      WRITE (6,105) X, Y, NWPN
  40  CONTINUE
  50  CONTINUE
  60  CONTINUE
      RETURN
 101 FORMAT('0',' TGT    NUMBER ',4X,'CPU',16X,'COVERAGE',58X,'BLDG',/,
     X '   NO     HITS    COVERAGE',12X,'R1',6X,'R2   PEOPLE    AGE  ',
     X    ' PARTS    AMMO    TRAP   MATERL ',7X,'NO',/,' ',7X,
     X    'TOT  OUT'  )
 102  FORMAT(' ',I3,3X,I3,2X,I3,3X,F6.2,10X,8(3X,F5.3),6X,I4,3X,4A4)
 103  FORMAT('0',15X,'HIT LOCATION AND WPN TYPE FOR SELECTED TARGETS'
     X,/,/)
```

```
  104   FORMAT(' ','    TARGET NUMBER',I4,'    X-DIM     Y-DIM    ',
       X 'WPN TYPE',/)
  105   FORMAT(' ',  21X,2I9,I7)
  106   FORMAT('1',25X,'TARGET HIT SUMMARY',/,/)
  107   FORMAT('1')
  108   FORMAT('0',25X,'TARGET HIT SUMMARY       TRIAL',I5)
  109   FORMAT('0', 25X,'** TARGET TYPE # ',I3,'  **',/)
        END
```

```
            SUBROUTINE STATIS
              IMPLICIT INTEGER *2 (A-Z)
              INTEGER *4  MSTAT
            INTEGER *4   MXITEM,NOPEO,NCAGE,NOPART,NOMUN,NCTRAP,NOMATL,NOPOL
            INTEGER *4   PEOPLE,AGE,PARTS,AMMO,TRAP,MATERL,POL,NOEQUI,NOPEOP
              REAL     *4 PK,FHIT,AHITS,ACOV,AVGREP,CLSD,FOPEN,SDC,SDH,
          X                STAT, TOTAL, TOTAL1, STA, SDREP, TRIAL, SUM1, SUM2
      C   THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
      //      DD    DSN=*.STEP1.COMMCN,DISP=SHR
      //      DD    *,DCB=BLKSIZE=800
            COMMON /STATS / STAT(500,17),MSTAT(8),NTRIAL,ITRIAL,NSTAT
            COMMON /CONTRL/ NREP,NPLOT,INW,INL,CHANGE,NSAVE,LIST,NJMEM
            CCMMON /STORES/ MXITEM,NOPEO,NOAGE,NOPART,NOMUN,NCTRAP,NOMATL,
          X  NOPOL,NOEQUI,NOPEOP,PECPLE(202,4),AGE(101,4),PARTS(401,4),
          X  AMMO(126,4),TRAP(26,4),MATERL(26,4),POL(10,4),STOCKS(1000,3),
          X  EQUIV(400)
            DIMENSION PK(8)
            WRITE (6,101)  NTRIAL
            AVGREP = 0.0
            DO 12   M = 1, MTT
            NN = 0
            IF (M .NE. SHELT)  GO TO 2
            SUM1 = 0.0
            SUM2 = 0.0
        2   CONTINUE
                                   TOTAL1 = 0.0                              ###
            DO 10   I = 1, NT
            IF (TGT(I,10) .NE. M)   GC TO 10
            NN = NN + 1
            IF (NN .EQ. 1)   WRITE (6,102) M
            FHIT = (STAT(I,3)/NTRIAL)*100.
            AHITS=  STAT(I,1)/NTRIAL
            TRIAL = NTRIAL
            SDH = (STAT(I,2) - TRIAL*AHITS*AHITS)/(TRIAL - 1.)
            SDH = SDH**(0.5)
            ACOV = STAT(I,4)/TRIAL
            SDC  = (STAT(I,5) - TRIAL*ACCV*ACOV)/(TRIAL - 1.)
            SDC = SDC**(0.5)
            DO 5  L = 1,8
        5   PK(L) = STAT(I,L+5)/TRIAL
            TOTAL = FHIT + AHITS + SDH + ACOV + SDC
            DC 6  L = 1,8
        6   TOTAL = TOTAL + PK(L)
            IF (TOTAL .EQ. 0.0)  GO TO 8
      C
              FLAG = 0
              IF ((TGT(I,14).GT.0).AND.(TGT(I,14).LT.1000))  FLAG = 1
      C
            IF (FLAG .EQ. 0)  WRITE (6,103)I,FHIT,AHITS,SDH,ACOV,SDC,
          X      (PK(L),L=1,8), (NAME(I,L),L=1,NAMES)
      C
            IF (FLAG .EQ. 1)  WRITE (6,106)I,FHIT,AHITS,SDH,ACOV,SDC,
          X     (PK(L),L=1,8), TGT(I,14), (NAME(I,L),L=1,NAMES)
      C
                                   TOTAL1 = TOTAL1 + PK(1)                   ###
        8   IF (M .NE. SHELT)  GO TC 10
            SUM1 = SUM1 + FHIT
            SUM2 = SUM2 + AHITS
       10   CCNTINUE
            IF ((M .NE. SHELT).OR.(NN .EQ. 0))  GC TC 11
            SUM1 = SUM1/NN
            SUM2 = SUM2/NN
            WRITE(6,108)  SUM1, SUM2
                                   GC TO 12                                  ###
       11    IF (NN .EQ. 0)  GO TC 12
                                   IF (M .LT. 5)  GO TO 12                   ###
                                   TOTAL1 = TOTAL1/NN                        ###
                                   WRITE(6,109) TOTAL1                       ###
       12   CCNTINUE
            IF (MCR .EQ. 0)   GO TO 50
            STA = NSTAT
            FOPEN = (STA/TRIAL)*100.
            NCLSD = NTRIAL - STA
            IF (NCLSD .EQ. 0)   GC TO 40
            CLSD = NCLSD
            AVGREP = MSTAT(1)/CLSD
            SDREP  = 0.0
            IF (NCLSD .GT. 1)
          X  SDREP = ((MSTAT(2)-CLSD*AVGREP*AVGREP)/(CLSD-1.))**(.5)
```

```
   40   CONTINUE
        WRITE (6,104)  FCPEN
        IF (NRFP .EQ. 1)   WRITE(6,105) AVGREP, SDREP
   50   CCNTINUE
C
C    USE SUBROUTINE RESTAT TC SUMMARIZE THE RESCURCE LOSS STATISTICS
C
        CALL RESTAT(PECPLE(1,1),NOPEC ,1,NTRIAL)
        CALL RESTAT(   AGE(1,1),NOAGE, 2,NTRIAL)
        CALL RESTAT( PARTS(1,1),NOPART,3,NTRIAL)
        CALL RESTAT(  AMMO(1,1),NOMUN, 4,NTRIAL)
        CALL RESTAT(  TRAP(1,1),NCTRAP,5,NTRIAL)
        CALL RESTAT(MATERL(1,1),NOMATL,6,NTRIAL)
        CALL RESTAT(   POL(1,1),NOPCL, 7,NTRIAL)
C
        RETURN
  101   FORMAT ('1', 30X,'TARGET DAMAGE STATISTICS FOR',I4,' TRIALS',/,/,
     X  ' TARGET  PERCENT  AVERAGE HITS  STD DEV  AVG CBU  STD '
     X,'DEV  BCMB COVERAGE',10X,' KILL PROBABILITIES',22X,'BLDG ',/,
     X  '  NO   ATTACKS HIT  PER ATTACK  OF HITS  CCVERAGE COVE',
     X 'RAGE  R1     R2    PEOPLE   AGE   PARTS  AMMO  TRAP',
     X  ' MATERL         BLDG NO/NAME ',//)
  102   FORMAT (' ',/, 40X, 'TARGET TYPE # ',I3,/)
  103   FORMAT(' ',I4,3X,F6.1,3X,F7.2,7X,F6.2,3X,F6.2,3X,F6.2,
     X  8F7.3, 6X, 4A4)
  104   FORMAT(' ',/, ' AT LEAST ONE MINIMUM RUNWAY SECTION WAS OPEN AFTE
     XR', F6.1,' PERCENT OF THE ATTACKS',/,/)
  105   FORMAT(' ','WHEN ALL RUNWAYS WERE CLOSED, ',F6.2,'(',F6.2,
     X') HOLES REQUIRED REPAIR, ON THE AVERAGE, TO PROVIDE',
     X  ' A MINIMUM RUNWAY',//)
  106   FORMAT(' ',I4,3X,F6.1,3X,F7.2,7X,F6.2,3X,F6.2,3X,F6.2,
     X  8F7.3, 2X,I3,1X, 4A4)
  108   FORMAT(' ', 8X,'-------',4X,'-------',/,' ', 8X,F6.2,3X,F6.2 )
  109   FORMAT(' ',55X,'--------',/,' ',54X,F6.2 )
        END
```

```
      SUBROUTINE CBU
        IMPLICIT INTEGER *2 (A-Z)
        INTEGER *4 NSTAT
        REAL    *4  PHI, S, C, T, CT, A0, A1, A2, A3, A4, A5, XT, YT,
     X          STAT, PK,  PSP, PST, PSCOV, TOTC, SIN, COS
C    THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
//      DD    DSN=*.STEP1.COMMON,DISP=SHR
//      DD    *,DCB=BLKSIZE=800
      COMMON /STATS / STAT(500,17),NSTAT(8),NTRIAL,ITRIAL,NSTAT
      COMMON /CBUHIT/ CBUHT(200,2), IR(50), KCBU, KPTI
      DIMENSION NCOV(16,16),ICOV(16,16),PSCOV(16,16,6),PST(6),
     X  X(4), Y(4), XX(4), YY(4), WD(10)
C     COMPUTE PATTERN DIAGONAL DIMENSION
      DO 2   NWPN = 1, 10
      WD(NWPN) = 0
      LTH = -EMD(NWPN,1,1)
      IF (LTH .LE. 0)   GO TO 2
      WID = EMD(NWPN,2,1)
      WD(NWPN) = ((LTH*LTH + WID*WID)**(.5))/2.
    2 CONTINUE
C   AT THIS POINT THE PROGRAM NOW CHECKS, TARGET BY TARGET, FOR
C   WHATEVER CBU'S MAY HAVE COVERED ANY PART OF THE TARGET.
          DO 200    L = 1, NT
C         *******************      FOR EACH TARGET
      INIT1 = 0
C    COMPUTE TARGET CENTER
      TCX=(TGT(L,1)+TGT(L,5))*0.5
      TCY=(TGT(L,2)+TGT(L,6))*0.5
C     TARGET DIAGONAL
      XA=TGT(L,1)-TGT(L,5)
      YA=TGT(L,2)-TGT(L,6)
      TD = 0.5*((XA*XA + YA*YA)**(0.5))
      NCBUHT = 0
      FLAG = 0
          DO 40    I = 1, NA
C         ******************      FOR EACH ATTACK
      IF (IR(I) .EQ. 1)   GO TO 40
      INIT2 = 0
      NWPN = ATT(I,9)
      LTH = -EMD(NWPN,1,1)
      IF (LTH .LE. 0)   GO TO 40
      WID = EMD(NWPN,2,1)
      TOT = WD(NWPN) + TD
      NS = ATT(I,7)
      INIT3 = 0
          DO 20    M = 1, NS
C         ******************      FOR EACH WEAPON
      NCBUHT = NCBUHT + 1
      XB = CBUHT(NCBUHT, 1)
      IF (XB .EQ. -10000)    GO TO 20
      YB = CBUHT(NCBUHT, 2)
C    DISTANCE BETWEEN TARGET AND PATTERN CENTERS
      D = ((XB-TCX)*(XB-TCX) + (YB-TCY)*(YB-TCY))**(.5)
C    TARGET CANNOT BE HIT IF D GREATER THAN TOT
      IF (D .GT. TOT)   GO TO 20
      FLAG = 1
      IF (INIT3 .GT. 0)   GO TO 16
      INIT3 = 1
      PHI = ATT(I,1)/57.3
      S = SIN(PHI)
      C = COS(PHI)
      SL = S*LTH
      SW = S*WID
      CL = C*LTH
      CW = C*WID
      DO 5   J = 1, 16
      DO 5   K = 1, 16
    5 NCOV(J,K) = 0
   16 CONTINUE
      XX(1) = XB - (SL+CW)/2.
      XX(2) = XX(1) +SL
      XX(3) = XX(2) + CW
      XX(4) = XX(1) + CW
      YY(1) = YB + (SW-CL)/2.
      YY(2) = YY(1) + CL
```

```
      YY(3) = YY(2) - SW
      YY(4) = YY(1) - SW
      IF (INIT2 .GT. 0)   GO TO 18
      INIT2 = 1
C   FIND WESTERLY CORNER
       ILX = 1
      DO 15   NN = 1,4
   15 IF (XX(NN) .LT. XX(ILX))   ILX = NN
      IF (S .EQ. 1.0)   ILX = 4
C   RENUMBER CORNERS SO THAT CORNER #1 IS THE MOST WESTERN
      IDIF = ILX - 1
C   COMPUTE AND ADJUST TAN AND COTAN AS REQUIRED
      IF ((S .EQ. 0.0) .OR. (C .EQ. 0.0))   GO TO 28
      IF ((ILX .EQ. 2) .OR. (ILX .EQ. 4))   GO TO 26
      T = S/C
      GO TO 27
   26 T = -C/S
   27 CT= 1./T
   28 CONTINUE
   18 DO 22   NN = 1,4
       NEW = NN - IDIF
      IF (NEW .LT. 1)   NEW = NEW + 4
       X(NEW) = XX(NN)
       Y(NEW) = YY(NN)
   22   IF (KTEST .GT. 4)  WRITE(6,1003)NEW,X(NEW),Y(NEW)
       IF (INIT1 .GT. 0)   GO TO 31
C   CREATE A 16-POINT GRID ON TARGET - USE MORE POINTS FOR LARGE TGTS
      INIT1 = 1
      NX0 = 8
      NY0 = 8
      IF (TGT(L,12) .GT. 250.)   NY0 = 16
      IF (TGT(L,12) .GT. 1000.)   NY0 = 32
      IF (TGT(L,13) .GT. 250.)   NX0 = 16
      IF (TGT(L,13) .GT. 1000.)   NX0 = 32
      NXT = NX0/2
      NYT = NY0/2
      DO 29   J = 1, NXT
      DO 29   K = 1, NYT
      ICOV(J,K) = 0
      DO 29  NP = 3,8
   29 PSCOV(J,K,(NP-2)) = 1.0
       NX1 = NX0 - 1
       NY1 = NY0 - 1
        X0 = NX0
        Y0 = NY0
        A0 = TGT(L,1)
        A1 = (TGT(L,7)-TGT(L,1))/X0
        A2 = (TGT(L,3)-TGT(L,1))/Y0
        A3 = TGT(L,2)
        A4 = (TGT(L,4)-TGT(L,2))/Y0
        A5 = (TGT(L,8)-TGT(L,2))/X0
   31 CONTINUE
C   TEST TO SEE IF TARGET CORNERS COVERED BY PATTERN
      NIN = 0
      NGIN = 0
       DO 10   NC = 1, 7, 2
      XT=TGT(L,NC)
      YT=TGT(L,NC+1)
      IF (KTEST .GT. 4)  WRITE(6,1001) XT,YT
      IF ((XT .LE. X(1)).OR.(XT .GE. X(3)))   GO TO 10
      IF ((YT .LE. Y(4)).OR.(YT .GE. Y(2)))   GO TO 10
       IF ((S .EQ. 0.) .OR. (C .EQ. 0.))   GO TO 9
      IF (YT .GE. (Y(1)+CT*(XT-X(1)))) GO TO 10
      IF (YT.GE.(Y(2)-T*(XT-X(2)))) GO TO 10
      IF (YT.LE.(Y(4)+CT*(XT-X(4)))) GO TO 10
      IF (YT.LE.(Y(1)-T*(XT-X(1)))) GO TO 10
    9 NIN = NIN + 1
   10 CONTINUE
      IF (KTEST .GT. 3)   WRITE(6,1002)  NIN
      IF (NIN .LT. 4)   GO TO 34
C   IF ALL CORNERS COVERED BY PATTERN, TARGET FULLY COVERED
      DO 32   J = 1, NXT
      DO 32   K = 1, NYT
   32 NCOV(J,K) = NCOV(J,K) + 1
      GO TO 33
```

```
C     IF PARTIALLY COVERED, ESTIMATE FRACTION THAT IS COVERED
   34   CONTINUE
          DO 30 NX = 1,NX1,2
          DO 30 NY = 1,NY1,2
          J = (NX+1)/2.
          K = (NY+1)/2.
C     GRID-POINT DIMENSIONS
         XT = A0 + NX*A1 + NY*A2
         YT = A3 + NY*A4 + NX*A5
C     CHECK IF WITHIN RECTANGLE ENCLOSING PATTERN THAT IS PARALLEL TO
C     AXES
        IF ((XT .LT. X(1)).OR.(XT .GT. X(3)))     GO TO 30
        IF ((YT .LT. Y(4)).OR.(YT .GT. Y(2)))     GO TO 30
         IF ((S .EQ. 0.) .OR. (C .EQ. 0.))   GO TO 35
C     CHECK IF POINT IS WITHIN ACTUAL CBU PATTERN
        IF (YT .GT.(Y(1)+CT*(XT-X(1))))    GO TO 30
        IF (YT .GT.(Y(2)- T*(XT-X(2))))    GO TO 30
        IF (YT .LT.(Y(4)+CT*(XT-X(4))))    GO TO 30
        IF (YT .LT.(Y(1)-T*(XT-X(1))))   GO TO 30
   35   NGIN = NGIN + 1
        NCOV(J,K) = NCOV(J,K) + 1
        IF (KTEST .GT. 5)  WRITE(6,1005) NX,NY,XT,YT,NGIN,NCOV(J,K)
   30   CONTINUE
   33   CONTINUE
        IF ((NIN + NGIN) .EQ. 0)   GO TO 20
C      RECORD ANY COVERAGE AS A 'HIT'
        NHIT(L) = NHIT(L) + 1
        NCBU(L) = NCBU(L) + 1
        IF (KTEST .GT. 4)  WRITE(6,1006) L, NHIT(L), NCBU(L)
        IF ((TGT(L,11) .LT. 1) .OR. (TGT(L,10) .EQ. 1.))   GO TO 130
        IF (NHIT(L) .GT. 25)   GO TO 130
        DO 120   J = 1, NST
        IF (MHIT(J) .EQ. 0) GO TO 130
        IF (MHIT(J) .NE. L) GO TO 120
        HIT (J,1,NHIT(L))=XB
        HIT (J,2,NHIT(L))=YB
        HIT (J,3,NHIT(L))=NWPN
        IF (NHIT(L) .EQ. 25)   WRITE (6,1007)  L, ITRIAL
        GO TO 130
  120   CONTINUE
  130   CONTINUE
        IF (TGT(L,10) .NE. 1)   GO TO 150
        IF (NHIT(L) .GT. 250)   GO TO 150
        DO 140   J = 1, 5
        IF (NRW(J) .EQ. 0)   GO TO 150
        IF (NRW(J) .NE. L)   GO TO 140
        HITR(J,1,NHIT(L)) = XB
        HITR(J,2,NHIT(L)) = YB
        HITR(J,3,NHIT(L)) = NWPN
        IF (NHIT(L) .EQ. 250)   WRITE(6,1009)   L, ITRIAL
  140   CONTINUE
  150   CONTINUE
C    *************************    RECYCLE FOR MORE WEAPONS
   20    CONTINUE
        IF (INIT2 .EQ. 0)   GO TO 40
        DO 160  NP = 3, 8
        PK = EMD(NWPN,TGT(L,10),NP)/100.
        PSP = 1. - PK
        DO 160   J = 1, NXT
        DO 160   K = 1, NYT
        IF (NP .GT.  3)  GO TO 160
        ICOV(J,K) = ICOV(J,K) + NCOV(J,K)
  160   PSCOV(J,K,NP-2) = PSCOV(J,K,NP-2)*PSP**NCOV(J,K)
C    *************************    RECYCLE FOR MORE ATTACKS
   40   CONTINUE
        IF (FLAG .EQ. 0)  GO TO 200
        TCOV = 0
        DO 165  NP = 3, 8
  165   PST(NP-2) = 0.0
        DO 170   J = 1, NXT
        DO 170   K = 1, NYT
        IF (ICOV(J,K) .GT. 0)    TCOV = TCOV + 1
        DO 170  NP = 3, 8
  170   PST(NP-2) = PST(NP-2) + (1. - PSCOV(J,K,NP-2))
        TOTC = NXT*NYT
        COV(L) = TCOV/TOTC
```

```
         DO 180  NP = 3, 8
     180 P(L,NP) = 1  - (1.-P(L,NP))*(1.-PST(NP-2)/TOTC)
         IF (KTEST .GT. 3) WRITE(6,1004)L,TCOV,TOTC,(PST(I),I=1,6),
      X                           COV(L),(P(L,I),I=1,8)
C    *************************     RECYCLE FOR MORE TARGETS
     200 CONTINUE
         RETURN
    1001  FORMAT(' ',' XT  ',F8.0,'   YT  ',F8.0)
    1002   FORMAT(' ', '    NIN ',I4)
    1003   FORMAT(' ', '    NEW ',I3,2I10)
    1004   FORMAT(' ','TGT',I4,' COV',I4,'  TOT',F6.0,' PST',
      X  6F6.3,'   COV',F6.4,' PK',6F6.3 )
    1005   FORMAT(' ','    NX ',I3,'     NY ',I3,2F10.0,2I10)
    1006   FORMAT(' ',' TGT ',I4,'   NHIT',I5,'   NCBU',I5)
    1007  FORMAT('0','***** ONLY 25 HITS WERE STORED FOR TARGET #',
      X  I3,' DURING TRIAL #',I4,' ****')
    1009  FORMAT('0','***** ONLY 250 HITS WERE STORED FOR TARGET',
      X ' #',I3,' DURING TRIAL #',I4,' *****')
         END
```

```
      SUBROUTINE PLOTHT(NH,NR,LTH,WID)
      IMPLICIT INTEGER *2 (A-Z)
      REAL *4  Y
      COMMON /HITS/ XN(250),YN(250),NZ(250)
      DIMENSION ICOL(130)
      DATA IBK / 1H /,IX/ 1H* /,IY/ 1H+ /,IS/ 1H- /,IE/ 1H' /
C     THIS ROUTINE PLOTS THE IMPACT POINTS (BUT NOT CRATERS) FOR
C     ALL HITS THAT HAVE BEEN STORED FOR A RUNWAY/TAXIWAY.  IT
C     WILL PLOT ALL HITS THAT AFFECT RUNWAY OPERATION UP TO 50 'FEET'
C     OF EITHER SIDE OF (UP TO) A 300 'FOOT' RUNWAY.  RUNWAY LENGTH
C     IS LIMITED TO 13000 'FEET'.
      IWID = WID/10 + 5
       LEN = LTH/100 + 1
       IF (LEN .GT. 129)  LEN = 129
       LU = LEN/10
       LI = 10*LU + 1
       LU = LU + 1
       DO 40   J = 1,40
               I = 41-J
       DO 10   N = 1,129
   10 ICOL(N) = IBK
      ICOL(1) = IE
       ICOL(LEN) = IE
      IF ((I .NE. 5) .AND. (I .NE. IWID))   GO TO 14
      DO 11   NS = 1,LEN
   11 ICOL(NS) = IS
      DO 12   NS = 1,LI,10
   12 ICOL(NS) = IE
   14 CONTINUE
      DO 20   N = 1,NH
      NY = YN(N)/10 + 5
      IF (NY .NE. I)   GO TO 20
      NX = XN(N)/100 + 1
      IF ((NX .LT. 1) .OR. (NX .GT. 129))   GO TO 20
      ICOL(NX) = IX
      IF (NZ(N) .EQ. 1)   ICOL(NX) = IY
   20 CONTINUE
      Y = I/5.
      LY=Y
      IF ((Y-LY) .NE. 0.0)   GO TO 30
       LYY = 5*LY - 5
      WRITE (6,101) LYY,(ICOL(M),M=1,129)
      GO TO 40
   30 WRITE (6,102)     (ICOL(M),M=1,129)
   40 CONTINUE
      WRITE (6,103)   ( I, I=1,12), NR
      RETURN
  101  FORMAT (' ',I2,129A1)
  102  FORMAT (' ',2X,129A1)
  103  FORMAT (' ',' 0', 12(8X,I2),/,
     X  40X,'   TENS BY THOUSANDS OF LENGTH UNITS ',/,/,
     X  40X, '  IMPACT POINTS ON RUNWAY NUMBER ',I2,/,
     X  37X, ' (* = POINT IMPACT WPN    + = CBU CENTROID)')
       END
```

```
      SUBROUTINE JMEMO(NJMEM,D,NA,KTEST)
      INTEGER *2 NJMEM, D, NA, KTEST
      INTEGER *4 EE
      DIMENSION E(9), EE(9), D(11)
C   THIS SUBROUTINE PROVIDES THE USER 'METHOD 0' AS OUTLINED IN THE
C   8-74 WANG 700 USERS'S MANUAL FOR JMEM OPEN-END METHODS.
C   THIS TRAJECTORY PROGRAM PERMITS AIDA USERS TO PRESCRIBE THE
C   ATTACK DATA AS IN JMEM.  THE SUBROUTINE LOGIC IS TAKEN DIRECTLY FROM
C   THE REFERENCED PUBLICATION AND USES NOTATION CLOSELY PARALLELING
C   THE ORIGINAL.  ONLY THE 'PATTERN RADIUS' COMPUTATION (USED WITH
C   ROCKEYE) IS OMITTED.
       NJMEM = NJMEM + 1
       NCNT = 0
      IF (NJMEM .EQ. 1)   WRITE (6,101)
      READ (5,102)   (EE(I),I=1,9)
      DO 16   I = 1,6
   16 E(I) = EE(I)
      DO 18   I = 7,9
   18 E(I) = EE(I)/1000.
      WRITE (6,103)  NA,  (D(I),I=1,6)
      WRITE (6,104)        (E(I),I=1,9)
      DO 20   I = 7,9
   20 D(I) = D(I+1)
      V1 = 1.688*E(1)
       V = V1
      TH = E(3)/57.3
      VX = V*COS(TH)
      VY = -V*SIN(TH)
      TR = (D(7) - 1.)*E(8) + (E(5)-1.)*E(9)
      DX = TR*VX/2.
      Y1 = E(2) - VY*TR
      YI = (Y1 + E(2))/2.
      Y  = YI
      TF = 0.0
      VF = 0.0
      THF=0.0
      IF (E(7) .GT. 500.)   GO TO 1
      YT = 0.0
      IF (E(7) .EQ. 0.0)   GO TO 2
      TPD=E(7)
      GO TO 3
    1 YT = E(7)
    2 TPD= 99.
    3 CONTINUE
      DG = 32.17/(E(4)*E(4))
    7 CONTINUE
      IF (KTEST .GT. 5)   WRITE (6,106) TF,Y,V
      NCNT = NCNT + 1
      IF (NCNT .GT. 2000)   GO TO 30
      NCNT2 = 0
      C = V/220. - 3.
      IF (C .GE. 3.)   C = 3.
      IF (C .LT. 1.)   C = 1.
     DD = C*DG*EXP(-Y/31000.)
      DT= 10./(DG*V*V)
      IF (DT .GT. 0.5)   DT = 0.5
      GO TO 13
    5 CONTINUE
      TF = TFO
      VY = VYO
      Y  = YO
      IF (KTEST .GT. 5)   WRITE(6,107)  DT, VY
      NCNT2 = NCNT2 + 1
      IF (NCNT2 .GT. 100)   GO TO 30
   13 CONTINUE
      TFO = TF
      TF  = TF + DT
      IF (TF .LT. TPD)   GO TO 4
      TF = TPD
      DT = TF - TFO
    4 CONTINUE
      VYO = VY
     VY  = VY*(1.-V*DT*DD) - 32.17*DT
      YO  = Y
      Y   = Y + DT*(VY+VYO)/2.
      IF ((Y - YT) .GE. -1.)   GO TO 6
```

```
      ZZ = VYO*V*DD + 32.17
      Z = VYO*VYO + 2.0*(YO-YT)*ZZ
      DT = (VYO + Z**(.5))/ZZ
      GO TO 5
    6 CONTINUE
      VXO = VX
      VX  = VX*(1.-V*DT*DD)
      DX = DX + DT*(VXO+VX)/2.
      V = (VX*VX + VY*VY)**(.5)
      IF (TF .EQ. TPD)   GO TO 12
      IF (E(7) .GT. 500.)   GO TO 11
      GO TO 10
   11 AID = ABS((Y-E(7)))
      IF (AID .LT. 1.)   GO TO 12
      IF (KTEST .GE. 5)   WRITE (6,109) AID, TF
   10 CONTINUE
      IF (KTEST .GE. 5)   WRITE(6,108) Y
      IF (Y .GE. 1.)   GO TO 7
      GO TO 15
   12 CONTINUE
      Z = -VY/V
      ZZ= (1.-Z*Z)**(.5)
      THF=ATAN(Z/ZZ)
      DG = 32.17/(E(5)*E(5))
      TPD= 99.
      YT = 0.0
      IF (E(7) .GE. 500.)   E(7) = 0.0
      GO TO 10
   15 Z  = -VY/V
      ZZ = (1.-Z*Z)**(.5)
      AI = ATAN(Z/ZZ)
      SR = (DX*DX + Y1*Y1)**(.5)
      IF (D(5) .EQ. 0.0)   GO TO 8
      BP = D(4)/1000.
      CP = D(5)/1000.
      GO TO 9
    8 BP = 0.000573*D(4)
      CP = BP
    9 CONTINUE
      U  = (E(6)*E(6)*TF*TF)
      D(4) = (U + (SR*SR*BP/Y1)**2.)**(.5)
      D(5)=(U+CP*CP*SR*SR)**(.5)
      SRT = SR/1000.
      D(8) = V1*TR*SIN(AI-TH)/SIN(AI)
      DISP = D(6)
      D(10) = DISP*SRT
      D(6)  = D(10)/Z
      AID = 57.3*AI
      THFD= 57.3*THF
      WRITE(6,105) TF, SR, AID, THFD
      RETURN
   30 WRITE (6,110)
      STOP
  101 FORMAT('1',20X,'JMEM FORMAT ATTACK DATA',/,/' ',
     X'ATT NO       HDG     X-MPI     Y-MPI     CEP(REP)     (DEP)     DIS
     XP',/' ',
     X'             SPEED  ALTITUDE       DIVE     TERM1     TERM2     WIN
     XD     TD/HF              T         TD',/,/)
  102 FORMAT( 6X, 9I6 )
  103 FORMAT ('0',I6,7I10 )
  104 FORMAT (' ',6X, 6F10.0, 3F10.3)
  105 FORMAT(' ',30X,'TF ',F5.2,' SEC   SR ',F7.0,' FEET   ',
     X 'IMPACT ANGLE ',F5.2,' DEG   (FUZING ANGLE ',F5.2,' DEG)')
  106 FORMAT(' ','  **  ',F6.3,' SEC  ',F8.0,' FEET  ',F8.1,' FT/SEC')
  107 FORMAT(' ',10X,'  DT ',F10.4,'    VERT VEL',F10.3)
  108 FORMAT(' ','ALT ',F10.2)
  109 FORMAT(' ','ALT DIFF',F10.2,'  TF ',F10.2)
  110 FORMAT(' ',/' ','LOOPING IN JMEMO.  CHECK INPUTS AND/OR TEST',
     X ' WITH KTEST = 6.')
        END
```

```
          SUBROUTINE EXPHIT
            IMPLICIT INTEGER *2 (A-Z)
            REAL     *4  PI,PHI,REL,S,C,CT,TH,CHI,SC,CC,AID1,AID2,DD,DR,
         X          STG,CTG,XT,YT,XX,YY,D,R,DEN,DEN1,EHIT1,TCOV,BLDGHT,PS,
         X    SEXP,    SIN,COS, SIGRS,SIGDS, SIGR,SIGD, C1,C2,C3, S1,S2,
         X          X, Y, F, EN, DLS, DIL1, DIL2, DIV, TSRS, TSDS, PSR
C        THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
//          DD     DSN=*.STEP1.COMMON,DISP=SHR
//          DD     *,DCB=BLKSIZE=800
          DIMENSION ID(8), EHIT1(100), PSR(3), TCOV(2), BLDGHT(2)
C       THIS ROUTINE ESTIMATES THE EXPECTED NUMBERS OF HITS FOR ATTACKS
C       WITH POINT-IMPACT WEAPONS ONLY.
          NCYCLE = 0
                IF (KTEST.GT.4)WRITE(6,1101)NTRIAL,NPRINT,MODE
          COUNT = 0
          PI = 3.14159
C
C       EXAMINE THE ATTACKS ONE AT A TIME
C
          DO 200   I = 1,NA
          NW = ATT(I,9)
          IF (EMD(NW,1,1) .GE. 0)    GO TO 5
          COUNT = COUNT + 1
          IF (COUNT .EQ. 1)   WRITE(6,1004)
          GO TO 200
    5     REL = WPNREL(NW)
          NV = NTYPE(NW)
          IF (NV .GT. 3)   NV = 3
          NFLAG = 0
          NS = ATT(I,7)
          LS = ATT(I,8)
          PHI= ATT(I,1)/57.3
          S  = SIN(PHI)
          C  = COS(PHI)
          IF ((ATT(I,1) .EQ. 0) .OR. (ATT(I,1) .EQ. 180))   GO TO 10
          CT = C/S
          GO TO 20
   10     NFLAG = 1
   20     CONTINUE
            NP = NS
            DLS = 0
            IF (NP .EQ. 1)   GO TO 24
            DLS = LS/(NP - 1)
   24     CONTINUE
           X = ATT(I,2) - S*LS/2.
           Y = ATT(I,3) - C*LS/2.
          SIGRS = 2.200*ATT(I,4)*ATT(I,4)  + ATT(I,6)*ATT(I,6)
          SIGDS = 2.200*ATT(I,5)*ATT(I,5)  + ATT(I,10)*ATT(I,10)
          TSRS = 2*SIGRS
          TSDS = 2*SIGDS
          SIGR = SIGRS**(.5)
          SIGD = SIGDS**(.5)
          F  = 1. / (SIGR*SIGD*6.2832)
          IF (KTEST .GT. 7)    WRITE(6,1008)I,LS,NP,DLS,X,Y,SIGR,
         X SIGD, F
          NR = 1
          EMDI = EMD(NW,SHELT,1)
   28     CONTINUE
C
C      FOR EACH TARGET, COMPUTE THE EXPECTED NUMBER OF HITS FOR ATTACK "I"
C
          DO 180   L = 1,NT2
          DEN = 0.0
          IF ((NR.EQ.2).AND.(EMDI.EQ.0))   GO TO 180
            TGTA = TGT2(L,12)
            TGTB = TGT2(L,13)
            IF (EMDI .NE. 0)   GO TO 30
            EN = 0
            GO TO 32
C
C       THE AVERAGE HIT DENSITY IS DETERMINED BY AVERAGING THE HIT DENSITY
C       OVER SEVERAL POINTS OF A RECTANGLE WHOSE DIMENSIONS ARE CHOSEN TO
C       MAKE ITS AREA EQUAL IN SIZE TO THE AREA WITHIN WHICH THE TARGET
C       COULD BE HIT, TAKING INTO ACCOUNT THE ROUNDED CORNERS OF THAT AREA.
C
   30       C1 = (TGTA + TGTB)/2.0
            C2 = 3.1415926/4.0
```

```
            C3 = - C1*ENDI - C2*ENDI*ENDI
            EN = (-C1 + (C1*C1 - 4*C3)**(0.5))/2.0
   32       TL1 = TGTA + 2*EN
            TL2 = TGTB + 2*EN
          TH = TGT2(L,9)/57.3
          STG = SIN(TH)
          CTG = COS(TH)
            U = 0.0
            V = 0.0
          IF (EN .EQ. 0.0)   GO TO 35
          U = (STG + CTG) * EN
          V = (STG - CTG) * EN
   35     CONTINUE
          XD(1) = TGT2(L,1) - U
          XD(2) = TGT2(L,2) + V
          XD(3) = TGT2(L,3) + V
          XD(4) = TGT2(L,4) + U
          XD(5) = TGT2(L,5) + U
          XD(6) = TGT2(L,6) - V
          XD(7) = TGT2(L,7) - V
          XD(8) = TGT2(L,8) - U
          CHI = TH - PHI
          SC = SIN(CHI)
          CC = COS(CHI)
          S1 = (SIGRS*CC*CC + SIGDS*SC*SC)**(.5)
          S2 = (SIGRS*SC*SC + SIGDS*CC*CC)**(.5)
          N1 = (4*TL1/S1) + 1
          N2 = (4*TL2/S2) + 1
          IF (KTEST .GT. 9)   WRITE(6,1009) N1, N2, S1, S2
          IF ((N1 + N2) .GT. 2)   GO TO 100
          DO 90   K = 1,7,2
          XT = XD(K)
          YT = XD(K+1)
          XX = X - XT
          IF (NFLAG .EQ. 1)   GO TO 40
          YY = YT - Y + CT*XX
           D = -YY*S
           R =  YY*C - XX/S
          GO TO 50
   40     D = XX
          R = (YT-Y)*C
   50     AID1 = D*D/TSDS
          IF (AID1 .GT. 12.)   GO TO 90
          DD = SEXP(-AID1)
          NCYCLE = NCYCLE + 1
          DO 80   N = 1,NP
          AID2 = R*R/TSRS
          IF (AID2 .GT. 12.)   GO TO 70
          DR = SEXP(-AID2)
          DEN = DEN + DR*DD
          NCYCLE = NCYCLE + 1
          IF (KTEST .GT. 10)  WRITE(6,1035)I,L,K,XT,YT,D,R,DD,DR,DEN
   70     R = R - DLS
   80     CONTINUE
   90     CONTINUE
          DEN = DEN/(4.*NP)
          GO TO 160
C     IF THE TARGET DIMENSIONS ARE SMALL (I.E. LESS THAN ONE-QUARTER
C     THE PROJECTION OF SIGMA PARALLEL TO THE TARGET EDGE) THE HIT
C     DENSITY IS TAKEN AS THE AVERAGE OF THE VALUES AT THE FOUR CORNERS.
C     IF IT IS LARGER, A GRID OF INTERNAL POINTS IS ESTABLISHED AND
C     THE HIT DENSITY IS TAKEN AS THE AVERAGE OVER THE INTERNAL POINTS.
  100     CONTINUE
          DIL1 = TL1/(N1+1)
          DIL2 = TL2/(N2+1)
            NN1 = N1 + 2
            NN2 = N2 + 2
          DO 150   M = 1,NN1
          DO 150   N = 1,NN2
          XT = XD(1) + (M-1)*DIL1*STG + (N-1)*DIL2*CTG
          YT = XD(2) + (M-1)*DIL1*CTG - (N-1)*DIL2*STG
          XX = X - XT
          IF (NFLAG .EQ. 1)   GO TO 110
          YY = YT - Y + CT*XX
          D = -YY*S
          R =  YY*C - XX/S
          GO TO 120
```

```
  110   D = XX
        R = (YI-Y)*C
  120   AID1 = D*D/TSDS
        IF (AID1 .GT. 12.)    GO TO 150
        DD = SEXP(-AID1)
        DIV = 0.0
        IF ((M.EQ.1).OR.(M.EQ.NN1))  DIV = 2.0
        IF ((N.EQ.1).OR.(N.EQ.NN2))  DIV = DIV + 2.0
        IF (DIV .NE. 0.0)  DD = DD/DIV
        NCYCLE = NCYCLE + 1
        DO 140   K = 1,NP
        AID2 = R*R/TSRS
        IF (AID2 .GT. 12.)    GO TO 130
        DR = SEXP(-AID2)
        DEN = DEN + DR*DD
        NCYCLE = NCYCLE + 1
        IF (KTEST .GT. 12) WRITE(6,1006)I,L,XT,YT,D,R,DD,DR,DEN
  130   R = R - DRS
  140   CONTINUE
  150   CONTINUE
          DEN = DEN/((1 + (N1 + N2) + N1*N2)*NP)
  160   CONTINUE
        DEN1 = DEN
        DEN = NS*F*REL*AIT(I,11)*DEN1/100.
        IF (KTEST .GT. 7)    WRITE (6,1007) I,L,DEN1,DEN
        EHIT1(L) = DEN*TL1*TL2
        COV2(L,NR) = COV2(L,NR) + EHIT1(L)
  170   CONTINUE
C
C     REPEAT FOR EACH TARGET
C
        IF ((NR .EQ. 2).OR.(NV .LT. 2))  GO TO 195
        PSR(1) = (100. - EMD(NW, SHELT, 12))/100.
        PSR(2) = (100. - EMD(NW, SHELT, 14))/100.
        DO 190   L = 1,NT2
        DO 190   N = 2,NV
  190   P2(L,N+1) = 1. - (1.- P2(L,N+1))*(PSR(N-1)**EHIT1(L))
C
        IF (KTEST .GT. 4) WRITE(6,1011) PSR,(EHIT1(K),K=1,6)
C
        PSR(1) = (100.-EMD(NW,SHELT,3))/100.
        PSR(2) = (100.-EMD(NW,SHELT,4))/100.
        PSR(3) = (100.-EMD(NW,SHELT,5))/100.
        DO 192  L = 1, NT2
        DO 192  N = 6, 8
  192   P2(L,N) = 1. - (1. - P2(L,N))*(PSR(N-5)**EHIT1(L))
C
        IF (KTEST .GT. 4) WRITE(6,1011) PSR,(EHIT1(K),K=1,6)
C
  195   CONTINUE
        NR = NR + 1
        EMDI = EMD(NW,SHELT,10)
        IF (NR .EQ. 2)  GO TO 28
        DO 198   L = 1,NT2
        PS = (100. - EMD(NW, SHELT, 13))/100.
  198   P2(L,5) = 1. - (1. - P2(L,5))*(PS**EHIT1(L))
  200   CONTINUE
C
C     REPEAT FOR EACH ATTACK
C
        DO 205  L = 1, NT2
        P2(L,1) = 1. - SEXP( -COV2(L,1) )
  205   P2(L,2) = 1. - SEXP( -COV2(L,2) )
        IF (NPRINT .GT. 2)  RETURN
        WRITE(6,1000) NCYCLE
        WRITE (6,1002)
        NN = 0
        TCOV(1) = 0.0
        TCOV(2) = 0.0
        BLDGHT(1) = 0.0
        BLDGHT(2) = 0.0
        DO 210   L = 1,NT2
        NN = NN + 1
        DO 208   I = 1,2
        TCOV(I) = TCOV(I) + COV2(L,I)
  208   BLDGHT(I) = BLDGHT(I) + (1. - SEXP(-COV2(L,I)))
```

```
      WRITE (6,1003)  L,COV2(L,1), COV2(L,2), (P2(L,J),J=1,8)
210   CONTINUE
      BLDGHT(1) = BLDGHT(1)/NN
      BLDGHT(2) = BLDGHT(2)/NN
      WRITE (6,1010)  TCOV(1), BLDGHT(1), TCOV(2), BLDGHT(2)
      RETURN
1000   FORMAT ('0', 25X, 'CYCLES', I7,//,' ',30X,'**  SHELTERS  ** ')
1002   FORMAT('0',10X,'TARGET',13X,'HITS',38X,'PROBABILITIES',/,12X,
     X 'NO',14X,'EXPECTED',14X,'HIT(R1) HIT(R2)  AC(R1) SHELTER  ',
     X   'AC(R2)  PEOPLE   AGE    PARTS' )
1003   FORMAT(' ',10X,I4,6X,F6.3, 9X,F6.3,6X,8F8.3 )
1004   FORMAT ('1','   COMPUTATION IGNORED THE CBU WEAPONS IN THE',
     X  ' EXPECTED VALUE CALCULATION' )
1005   FORMAT(' ','CD   ATT ',I3,'   TGT ',I3,'   COR ',I2,
     X 4F6.0, 3E12.4 )
1006   FORMAT (' ','IP   ATT ',I3,'   TGT ',I3,4F6.0,3E12.5)
1007   FORMAT(' ','       ATT ',I3,'   TGT ',I3,'   AVG DEN ',
     X   F8.5,'   NOR DEN ',F14.10,//)
1008   FORMAT (' ','ATT ',I3,2I8, F8.2, 4F8.1, E12.5)
1009   FORMAT(' ','N1  ',I4,'    N2  ',I4,6X,2F10.1)
1010   FORMAT (' ',20X,'-------',/,19X,F8.3,'(',F5.3,')',
     X                                  F8.3,'(',F5.3,')' )
1011  FORMAT(' ',20X,'PSR',10F8.4 )
1101          FORMAT(' ',' ****** EXPHIT ', 3I8 )
      END
```

```
      FUNCTION SEXP(X)
      IF (X .LT. -0.025)   GO TO 10
      SEXP = 1.+X
      RETURN
10    SEXP = EXP(X)
      RETURN
      END
```

```
            SUBROUTINE STORE( ITEM, NCARDS)
            IMPLICIT INTEGER *2 (A-Z)
            REAL   *4  A
      C   THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
      //       DD     DSN=*.STEP1.COMMON,DISP=SHR
      //       DD     *,DCB=BLKSIZE=800
            INTEGER *4   CLASS, TYPE
            INTEGER *4   MXITEM,NOPEO,NOAGE,NOPART,NOMUN,NOTRAP,NOMATL,NOPOL
            INTEGER *4   PEOPLE,AGE,PARTS,AMMO,TRAP,MATERL,POL,NOEQUI,NOPEOP
            COMMON /STORES/ MXITEM,NOPEO,NOAGE,NOPART,NOMUN,NOTRAP,NOMATL,
           X  NOPOL,NOEQUI,NOPEOP,PEOPLE(202,4),AGE(101,4),PARTS(401,4),
           X  AMMO(126,4),TRAP(26,4),MATERL(26,4),POL(10,4),STOCKS(1000,3),
           X  EQUIV(400)
            DIMENSION A(5), D(15)
            FLAG = 0
            CARD = 1
         10 READ(5,1001) D(1),D(2),A(1),D(4),D(5),A(2),D(7),D(8),A(3),
           X              D(10),D(11),A(4),D(13),D(14),A(5)
       1001 FORMAT( 5X, 5(3X,I2,I5,F5.0)  )
            DO 15  N = 1,5
            IF (D(3*N-2) .EQ. 0)   GO TO 16
         15 D(3*N) = 10.0*(A(N) + 0.0001)
            MAX = 15
            GO TO 17
         16 MAX = 3*(N-1)
         17 IF (NPRINT.LE.2)  WRITE(6,1002)  (D(N),N=1,MAX)
       1002 FORMAT(' ',25X,5('  C',I3,2I5,4X) )
            N = 1
         20 CLASS = D(N)
            TYPE  = D(N+1) + 1
            STOCKS(ITEM,1) = NT
            STOCKS(ITEM,2) = D(N+2)
            GO TO 50
      C     **********
            ENTRY FILE(ITEM,CLASS,TYPE)
      C     **********
            FLAG = 1
            N1000 = 1000
            IF (NPRINT.LE.2)WRITE(6,1002) CLASS, TYPE, N1000
            TYPE = TYPE + 1
         50 GO TO (100,200,300,400,500,600,700), CLASS
      C
        100   IF (TYPE .LT. 1000)  GO TO 110
              TYPE = TYPE - 999 + NOPEOP
        110 NF = PEOPLE(TYPE,1)
            IF (NF .NE. 0)  GO TO 800
            PEOPLE(TYPE,1) = ITEM
            GO TO 1000
        200 NF = AGE(TYPE,1)
            IF (NF .NE. 0)  GO TO 800
            AGE(TYPE,1) = ITEM
            GO TO 1000
        300 NF = PARTS(TYPE,1)
            IF (NF .NE. 0)  GO TO 800
            PARTS(TYPE,1) = ITEM
            GO TO 1000
        400 NF = AMMO(TYPE,1)
            IF (NF .NE. 0)  GO TO 800
            AMMO(TYPE,1) = ITEM
            GO TO 1000
        500 NF = TRAP(TYPE,1)
            IF (NF .NE. 0)  GO TO 800
            TRAP(TYPE,1) = ITEM
            GO TO 1000
        600 NF = MATERL(TYPE,1)
            IF (NF .NE. 0)  GO TO 800
            MATERL(TYPE,1) = ITEM
            GO TO 1000
        700 NF = POL(TYPE,1)
            IF (NF .NE. 0)  GO TO 800
            POL(TYPE,1) = ITEM
            GO TO 1000
      C
        800 NX = STOCKS(NF,3)
            IF (NX .EQ. 0)  GO TO 900
            NF = NX
            GO TO 800
```

```
  900   STOCKS(NF,3) = ITEM
 1000   STOCKS(ITEM,3) = 0
C
        IF(KTEST.GT.4)WRITE(6,1101)ITEM,CLASS,TYPE,(STOCKS(ITEM,K),K=1,3)
  1101  FORMAT(' ',' STORE ',10I8 )
C
C
        ITEM = ITEM + 1
        IF (FLAG .EQ. 1)   RETURN
        N = N + 3
        IF ((N .NE. 16) .AND. (D(N) .GT. 0))  GO TO 20
         IF (CARD .EQ. NCARDS)  RETURN
        CARD = CARD + 1
        GO TO 10
        END
```

```
      SUBROUTINE SAVER(FLAG, KTEST, MAXLOC )
      IMPLICIT INTEGER *2 (A-Z)
      INTEGER *4  FLAG, KIND
C
     INTEGER *4   MXITEM,NOPEO,NOAGE,NOPART,NOMUN,NOTRAP,NOMATL,NOPOL
     INTEGER *4   PEOPLE,AGE,PARTS,AMMO,TRAP,MATERL,POL,NOEQUI,NOPEOP
     COMMON /STORES/ MXITEM,NOPEO,NOAGE,NOPART,NOMUN,NOTRAP,NOMATL,
    X NOPOL,NOEQUI,NOPEOP,PEOPLE(202,4),AGE(101,4),PARTS(401,4),
    X AMMO(126,4),TRAP(26,4),MATERL(26,4),POL(10,4),STOCKS(1000,3),
    X EQUIV(400)
     COMMON / ZERO / EQUI(7)
C
      IF (FLAG .EQ. 1)  GO TO 10
      LOC = 1
      CLASS = 0
      LAST = 0
      DO  5  I = 1,7
    5 EQUI(I) = 0
      RETURN
C
   10 EQUIV(LOC) = 0
      MAXLOC = LOC
      RETURN
C     **********
      ENTRY SAVE(CLAS, TYP, KIND)
C     **********
      TYPE = TYP + 1
C
       IF (KTEST .GT. 5) WRITE(6,1101) LOC,CLAS,CLASS,TYPE,LAST,KIND
 1101  FORMAT(' ',' SAVER ', 8I8 )
C
      IF (CLAS .NE. CLASS)  GO TO 50
      IF (TYPE .EQ. LAST)   GO TO 800
   50 EQUIV(LOC) = 0
      LOC = LOC + 1
      LAST = TYPE
      CLASS = CLAS
C
      GO TO (100,200,300,400,500,600), CLASS
C
  100 IF (TYPE .LT. 1000)  GO TO 110
      TYPE = TYPE - 999 + NOPEOP
  110 PEOPLE(TYPE,4) = LOC
      GO TO 800
  200 AGE(TYPE,4) = LOC
      GO TO 800
  300 PARTS(TYPE,4) = LOC
      GO TO 800
  400 AMMO(TYPE,4) = LOC
      GO TO 800
  500 TRAP(TYPE,4) = LOC
      GO TO 800
  600 MATERL(TYPE,4) = LOC
      GO TO 800
C
  800 EQUIV(LOC) = KIND
      IF (KIND .EQ. 0)  EQUI(CLASS) = 1
      LOC = LOC + 1
      IF (LOC .LT. NOEQUI)  RETURN
      WRITE(6,1001) CLAS, TYP, KIND
 1001 FORMAT(' ','THE EQUIVALENCE ARRAY IS TOO SMALL; OVERFLOWED ',
     X           'AT ',3I8 )
      STOP
      END
```

```
      SUBROUTINE DAMAGE
      IMPLICIT INTEGER *2 (A-Z)
      INTEGER *4  NBASE1, NBASE2, MSTAT, AREA, TOTAL, KAREA, DISK, MXNT
      INTEGER *4   MXITEM,NOPEO,NOAGE,NOPART,NOMUN,NOTRAP,NOMATL,NOPOL
      INTEGER *4   PEOPLE,AGE,PARTS,AMMO,TRAP,MATERL,PCL,NOEQUI,NOPEOP
      REAL    *4   PK, STAT, TOTCO1,TOTCO2,TOTCO3,TOTCO4,TOTCO5,TOTCO6
C    THE FOLLOWING JCL INSERTS TSARINA'S CCMMON "BASIC"
//        DD     DSN=*.STEPT.COMMON,DISP=SHR
//        DC     *,DCB=BLKSIZE=800
      COMMON /STORES/ MXITEM,NOPEO,NOAGE,NCPART,NOMUN,NOTRAP,NOMATL,
     X  NOPOL,NOEQUI,NOPEOP,PEOPLE(202,4),AGE(101,4),PARTS(401,4),
     X  AMMO(126,4),TRAP(26,4),MATERL(26,4),POL(10,4),STOCKS(1000,3),
     X  EQUIV(400)
      COMMON /STATS / STAT(500,17),MSTAT(8),NTRIAL,ITRIAL,NSTAT
      COMMON /OUTPUT/ TSAR, PUNCH, NBASE1, NBASE2
      DIMENSION DATA(15)
C
      IF ((NPRINT .GE. 3).AND.(ITRIAL .EQ. 1))  WRITE(6,2222)
 2222 FORMAT('1')
C
C  *****************************************************************
C
C    THE OUTPUTS FOR TSAR MAY BE PRINTEL, PUNCHED ON CARDS AND FILED
C    DIRECTLY ON DISK FOR PRODUCTION RUNS.  THE 2NC ANC 3RD CPTIONS
C    ARE CONTROLLED BY THE VARIABLES PUNCH AND TSAR, RESPECTIVELY,
C    AND ARE EXERCISED WHEN THE VARIABLES ARE INITIALIZED TO UNITY.
C
C  *****************************************************************
        MXNT = NTM
        DISK = ITRIAL + 20
        CARDTY = 40
        JBASE   = BASE
      DC 10  K = 1,15
   10 DATA(K) = 0
      DATA(1) = DAY
      DATA(2) = HOUR
      DATA(3) = MINUTE
C
      IF (PDAM.GT.1)  WRITE(6,7777) CARDTY,JBASE,(DATA(K),K=1,15)
      IF (TSAR.EQ.1)  WRITE(DISK)   CARDTY,JBASE,(DATA(K),K=1,15)
      IF (PUNCH.EQ.1) WRITE(7,8886) ITRIAL,JBASE,NBASE1,NBASE2,DAY,HOUR
      IF (PUNCH.EQ.1) WRITE(7,8888) CARDTY,JBASE,(DATA(K),K=1,15)
C
C   RECORD DAMAGE SUSTAINED BY VARIOUS RESOURCE STORAGE AREAS
C
      CALL IREPT(MXITEM, MXNT, NOEQUI, STOCKS(1,1), P(1,1), TGT(1,1),
     X                      EQUIV(1), NOPEOP, PDAM, KTEST )
C
       CALL REPORT( 1, PEOPLE(1,1), NOPEO,  3)
       CALL REPORT( 2, AGE(1,1),    NCAGE,  4)
       CALL REPORT( 3, PARTS(1,1),  NCPART, 5)
       CALL REPORT( 4, AMMO(1,1),   NOMUN,  6)
       CALL REPORT( 5, TRAP(1,1),   NCTRAP, 7)
       CALL REPORT( 6, MATERL(1,1), NCMATL, 8)
       CALL REPORT( 7, POL(1,1),       10,  6)
C
C  PREPARE TSAR INPUT DATA FOR THE SHOPS
C
      DO 20  K = 1,3
   20 DATA(K) = 0
      JBASE = 0
      N = 3
      DO 40 SHOP = 1,30
      I = FACLTY(SHOP)
      IF (I.EQ.0)  GO TO 40
      IF (P(I,1) .EQ. 0.0)  GO TO 40
      IF (N .EQ. 15)  N = 3
      DATA(N+1) = 9
      DATA(N+2) = SHOP
      DATA(N+3) = 100*(P(I,1) + 0.005)
      DO 30  K = 4,6
   30 DATA(N+K) = 100*(P(I,(K-1)) + 0.005)
      N = N + 6
C
      IF ((N.EQ.15).AND.(PDAM.GT.1))WRITE (6,7777) CARDTY,JBASE,
     X    (DATA(K),K=1,15)
      IF ((TSAR.EQ.1).AND.(N.EQ.15)) WRITE(DISK) CARDTY,JBASE,
     X                      (DATA(K),K=1,15)
      IF ((PUNCH.EQ.1).AND.(N.EQ.15)) WRITE(7,8888) CARDTY,JBASE,
     X                      (DATA(K),K=1,15)
C
   40 CONTINUE
      IF (N-9)  70, 50, 70
```

```
   50   DO 60  K = 10,15
   60   DATA(K) = 0
C
        IF (PDAM.GT.1)  WRITE(6,7777) CARDTY,JBASE,(DATA(K),K= 1,15)
        IF (DSAR.EQ.1)  WRITE(DISK)    CARDTY,JBASE,(DATA(K),K=1,15)
        IF (PUNCH.EQ.1) WRITE(7,8888) CARDTY,JBASE,(DATA(K),K=1,15)
C
   70   CONTINUE
C
C     BEGIN TO ESTIMATE THE FRACTION OF SHELTERED AND
C     OF PARTIALLY EXPOSED AIRCRAFT THAT ARE LOST.
C
        TOTC1 = 0
        TOTC2 = 0
        TOTC3 = 0
        TOTCO1 = 0.0
        TOTCO2 = 0.0
        TOTCO3 = 0.0
        TOTCO4 = 0.0
        TOTCO5 = 0.0
        TOTCO6 = 0.0
C
C     CHECK WHETHER SHELTERS ARE HANDLED WITH TGT2 CARDS OR INDIVIDUALLY
C
        IF ((NT2 .EQ. 0).AND.(NSM .EQ. 0))   GO TO 215
        MAXS = NT2
        IF (NT2 .EQ. 0)  MAXS = NSM
C
        DO 210  K = 1,MAXS
        TOTCO1 = TOTCO1 + P2(K,3)
        TOTCO2 = TOTCO2 + P2(K,4)
        TOTCO3 = TOTCO3 + P2(K,5)
        IF (NT2 .EQ. 0)  GO TO 200
C
        TOTCO4 = TOTCO4 + P2(K,6)
        TOTCO5 = TOTCO5 + P2(K,7)
        TOTCO6 = TOTCO6 + P2(K,8)
        GO TO 210
C
  200   TOTCO4 = TOTCO4 + P(SHEL(K),3)
        TOTCO5 = TOTCO5 + P(SHEL(K),4)
        TOTCO6 = TOTCO6 + P(SHEL(K),5)
  210   CONTINUE
C
                    IF (KTEST.GT.3)  WRITE(6,1102) TOTCO1, TOTCO2, TOTCO3,
     X              TOTCO4, TOTCO5, TOTCO6, ACLOSS
 1102               FORMAT(' ','DAMAGE-A', 6F8.4, I8 )
C
        TOTC1 = (100*(TOTCO1 + 0.005))/MAXS
        TOTC2 = (100*(TOTCO2 + 0.005))/MAXS
        TOTC3 = (100*(TOTCO3 + 0.005))/MAXS
C
C        ESTIMATE THE PERCENTAGE LOSS TO EXPOSED AIRCRAFT AS A FRACTION
C        OF THE RAMP SPACE THAT IS COVERED BY GP OR CPU EFFECTS.
C
  215   TOTAL = 0
        KAREA = 0
        UNSHEL = 0
        TOTHIT = 0
C
        DO 230  I = 1,NT
        IF (TGT(I,10) .NE. RAMPS)  GO TO 220
        AREA = TGT(I,12)*TGT(I,13)
        PK = 1. - (1.-P(I,5))
        TOTAL = TOTAL + AREA
        KAREA = KAREA + PK*AREA
  220   CONTINUE
        IF (TOTAL .GT. 0)  UNSHEL =  (100*KAREA)/TOTAL
        IF (TGT(I,10) .NE. TXWYS)  GO TO 230
        TOTHIT = TOTHIT + NHIT(I)
  230   CONTINUE
C
C        RECORD THE REPAIRS ON THE LEAST DAMAGED RUNWAY AND
C        THE TOTAL NUMBER OF HITS ON THE TAXIWAYS.
C
        IF ((REPAIR + TOTHIT) .EQ. 0)  GO TO 250
        DATA(4) = 9
        DATA(5) = NOPAC
        DATA(6) = REPAIR
C
        DO 240  K = 7, 9
  240   DATA(K) = 0
C
```

```
            DATA(10)  = 9
            DATA(11)  = NOFAC - 1
            DATA(12)  = TOTHIT
      C
            IF (PDAM.GT.1)  WRITE(6,7777) CARDTY,JBASE,(DATA(K),K=1,12)
            IF (TSAR.EQ.1)  WRITE(DISK)   CARDTY,JBASE,(DATA(K),K=1,12)
            IF (PUNCH.EQ.1) WRITE(7,8888) CARDTY,JBASE,(DATA(K),K=1,12)
      C
       250  CONTINUE
      C
      C   STORE THE SHELTER AND AIRCRAFT DAMAGE DATA USING
      C   THE SPECIAL CLASS #10 ENTRIES FOR TSAR.
      C
            IF ((TOTC1+TOTC2+TOTC3+UNSHEL) .EQ. 0)  GO TO 310
            DATA(4) = 10
            DATA(5) = 1
            DATA(6) = TOTC2
            DATA(7) = TOTC3
            DATA(8) = UNSHEL
            DATA(9) = 0
      C
            DATA(10) = 10
            DATA(11) = 2
            DATA(12) = ACLOSS
      C
            IF (PDAM.GT.1)  WRITE(6,7777) CARDTY,JBASE,(DATA(K),K=1,12)
            IF (TSAR.EQ.1)  WRITE(DISK)   CARDTY,JBASE,(DATA(K),K=1,12)
            IF (PUNCH.EQ.1) WRITE(7,8888) CARDTY,JBASE,(DATA(K),K=1,12)
      C
      C          STORE THE EXPECTED DAMAGE TO AIRCRAFT
      C
            DATA(4) = 8
            DATA(5) = 0
            DATA(6) = TOTC1
      C
      C   STORE THE RESOURCE LOSS RATES EXPECTED WHEN AIRCRAFT ARE DAMAGED
      C
      C   ESTIMATE THE LIKELIHOOD THAT PEOPLE, AGE AND SPARE PARTS
      C   ARE LOST WHEN AN AIRCRAFT IS DAMAGED AS THE RATIO OF THE
      C   PROBABILITY THAT THE RESOURCE IS LOST, DIVIDED BY THE
      C   PROBABILITY THAT THE AIRCRAFT IS LOST.
      C
      C      DATA ARE ENTERED IN THE ORDER:  PEOPLE, AGE AND PARTS
      C
                    PEODAM = 0
                    AGEDAM = 0
                    PRTDAM = 0
      C
                IF (TOTCO1 .EQ. 0.0)  GO TO 290
      C
            PEODAM = ((100.*TOTCO4)/TOTCO1 + 0.5)
            AGEDAM = ((100.*TOTCO5)/TOTCO1 + 0.5)
            PRTDAM = ((100.*TOTCO6)/TOTCO1 + 0.5)
      C
            GO TO 300
      C
       290  CONTINUE
      C
      C   IF THE PK AGAINST AIRCRAFT IN SHELTERS IS ZERO, USE THE
      C   FOLLOWING RELATIONSHIPS TO PROVIDE A PROXY FOR THE LOSS
      C   RATES FOR PERSONNEL, AGE AND PARTS ASSOCIATED WITH
      C   AIRCRAFT THAT ARE IN THE OPEN.
      C
            PEODAM = (100*EMD(1,SHELT,3))/EMD(1,SHELT,13)
            AGEDAM = (100*EMD(1,SHELT,4))/EMD(1,SHELT,13)
            AGEPRT = (100*EMD(1,SHELT,5))/EMD(1,SHELT,13)
      C
       300  CONTINUE
      C
            IF (PEODAM .GT. 100)  PEODAM = 100
            IF (AGEDAM .GT. 100)  AGEDAM = 100
            IF (PRTDAM .GT. 100)  PRTDAM = 100
      C
            DATA(7) = PEODAM
            DATA(8) = AGEDAM
            DATA(9) = PRTDAM
      C
            IF (PDAM.GT.1)  WRITE(6,7777) CARDTY,JBASE,(DATA(K),K=1,9)
            IF (TSAR.EQ.1)  WRITE(DISK)   CARDTY,JBASE,(DATA(K),K=1,9)
            IF (PUNCH.EQ.1) WRITE(7,8888) CARDTY,JBASE,(DATA(K),K=1,9)
      C
       310    CONTINUE
      C
```

```
      IF (LAST .EQ. 0)  RETURN
        CARDTY = 0
      IF (PDAM.GT.1)  WRITE(6,7777) CARDTY
      IF (ISAR.EQ.1)  WRITE(DISK)   CARDTY
      IF (PUNCH.EQ.1) WRITE(7,8888) CARDTY
      RETURN
 7777 FORMAT(' ',I2,I4, 15I7)
 8888 FORMAT('******      TRIAL',I4,'    BASE ',I3,4X,'NAME  ',2A4,4X,
     X        '    DAY HR', 2I4 )
 8888 FORMAT(I2,I3,15I5)
      END
```

```
      SUBROUTINE REPORT(CLASS, A, MA, NR)
      IMPLICIT INTEGER *2  (A-Z)
      INTEGER *4 MSTAT,CLASS,MXNT,MAX,NOEQUI,NOPEOP,A,MA,NR,DISK
      INTEGER *4 NBASE1,NBASE2
      REAL     *4    STAT,   P,  FRAC,  TOT,  TOT1
      COMMON /STATS / STAT(500,17),MSTAT(8),NTRIAL,ITRIAL,NSTAT
      COMMON /OUTPUT/ TSAR, PUNCH, NBASE1, NBASE2
      COMMON / ZERO / EQUI(7)
      DIMENSION  STOCKS(MAX,3),A(MA,4),DATA(15),P(MXNT,8),
     X               TGT(MXNT,14), EQUIV(NOEQUI)
C  *******************************************************************
C
C     THE OUTPUTS FOR TSAR MAY BE PRINTED, PUNCHED CN CARDS AND FILED
C     DIRECTLY CN DISK FOR PRODUCTION RUNS.  THE 2ND AND 3RD OPTIONS
C     ARE CONTRCLLED BY THE VARIABLES PUNCH AND TSAR, RESPECTIVELY,
C     ANC ARE EXERCISED WHEN THE VARIABLES ARE INITIALIZED TO UNITY.
C
C     THE PRINT OPTIONS ARE CCNTROLLED BY THE CONTROL VARIABLE "PDAM"
C
C  *******************************************************************
      DISK = ITRIAL + 20
      SKIP = 0
      IF ((CLASS.NE.1).AND.(A(1,1).EQ.0).AND.(EQUI(CLASS).EQ.0)) SKIP=1
      CARDTY = 40
       JBASE = 0
      M = 3
      DO 10  K = 1,15
 10   CATA(K) = 0
 20   IF (MA .EQ. 1)  GO TO 160
      DO 150  I = 2, MA
      IF (CLASS .NE. 1)  GO TO 30
      IF (I .EQ. ALL)  GO TC 150
 30   ITEM = A(I,1)
      TOT = 0.0
      IF (ITEM .NE. 0)  GO TO 40
      IF (A(I,4) .EQ. 0)  GO TO 150
      GO TO 60
 40   NT = STOCKS(ITEM,1)
      FRAC = STOCKS(ITEM,2)/10.0
      TOT  =  TOT + FRAC*P(NT,NR)
      ITEM = STOCKS(ITEM,3)
      IF (ITEM .GT. 0)  GO TO 40
      IF (TOT .EQ. 0.0)  GO TC 50
      A(I,2) = A(I,2) + (10*TCT)
      A(I,3) = A(I,3) + (100*TCT*TOT)
 50   TYPE = I - 1
      TOT1 = TOT/100.
      TTYPE = TYPE
      IF ((CLASS .EQ. 1).ANC.(TYPE.GT.NOPEOP)) TTYPE = 999+TYPE-NCPEOP
      IF((PDAM.EQ.1).OR.(PDAM.EQ.3)) WRITE(6,7766)  CLASS, TTYPE, TOT1
C
              IF(KTEST.GT.1) WRITE(6,1101) I,CLASS,A(I,4),TOT
1101          FCRMAT(' ',' REPORT', 3I10, F10.4)
C
C   A REPORT OF ZERO LOSSES MAY BE REQUIRED FOR TSAR
C
         IF (A(I,4) .EQ. 0)  GC TO 100
         IF ((SKIP .EQ. 1).AND.(TOT .EQ. 0.0))  GO TC 150
C
C   WHEN THE RESOURCE TYPE SPECIFIED IN THE TSARINA TARGET ARRAYS
C   IS A COMPCSITE CF SEVERAL TYPES USED IN TSAR, REPCRT THE DAMAGE
C   TO TSAR FOR EACH TYPE.
C
  60     LOC = A(I,4)
  70     KIND = EQUIV(LOC)
         IF (KINC .LT. 0)  GC TO 150
  75       IF (KTEST.GT.5) WRITE(6,1101) I, CLASS, KIND, TOT
         DATA(M+1) = CLASS
         DATA(M+2) = KIND
         DATA(M+3) = (TCT + 0.5)
         M = M + 6
         IF (M .NE. 15)  GO TO 90
        IF (PDAM .GT. 1) WRITE(6,7777) CARDTY,JBASE, (DATA(K),K=1,15)
        IF (TSAR.EQ.1)   WRITE(DISK)   CARDTY,JBASE, (DATA(K),K=1,15)
        IF (PUNCH.EQ.1)  WRITE(7,8888) CARDTY,JBASE, (DATA(K),K=1,15)
         DO 80  K = 1,15
  80     DATA(K) = 0
         M = 3
  90     LOC = LOC + 1
```

```
              KIND = EQUIV(LOC)
              IF (KIND .EQ. 0)  GO TO 150
              GO TO 75
      C
       100    CONTINUE
              IF (CLASS .NE. 1)  GO TO 105
              IF (I .GT. ALL)    GO TO 110
       105    IF ((TOT .EQ. 0.0).AND.(A(1,1) .EQ. 0.0))  GO TO 150
              GO TO 115
       110    IF ((TOT .EQ. 0.0).AND.(A(ALL,1).EQ.0.0))  GO TO 150
       115    CONTINUE
      C
      C     ONLY ONE TYPE OF POL IS PRESENTLY RECOGNIZED IN TSAR:  TYPE #1.
      C     DAMAGE DATA ON OTHER POL TYPES IS NOT STORED FOR USE BY TSAR.
      C
            IF ((CLASS .EQ. 7).AND.(TYPE .NE. 1))  GO TO 120
            DATA(M+1) = CLASS
            DATA(M+2) = TYPE
            IF ((CLASS.EQ.1).AND.(I.GT.MAXPEO)) DATA(M+2)= 1000 + I - ALL
            DATA(M+3) = (TOT + 0.5)
            M = M + 6
       120  IF (M .NE. 15)  GO TO 150
            IF (PDAM .GT. 1)  WRITE(6,7777)  CARDTY,JBASE,(DATA(K),K=1,15)
            IF (TSAR.EQ.1)    WRITE(DISK)    CARDTY,JBASE,(DATA(K),K=1,15)
            IF (PUNCH.EQ.1)   WRITE(7,8888)  CARDTY,JBASE,(DATA(K),K=1,15)
       130  DO 140  K = 1,15
       140  DATA(K) = 0
            M = 3
       150  CONTINUE
            IF (M .NE. 9)  GO TO 160
            IF (PDAM .GT. 1)   WRITE(6,7777) CARDTY,JBASE, (DATA(K),K=1,9)
            IF (TSAR .EQ. 1)   WRITE(DISK)   CARDTY,JBASE, (DATA(K),K=1,9)
            IF (PUNCH .EQ. 1)  WRITE(7,8888) CARDTY,JBASE, (DATA(K),K=1,9)
       160  CONTINUE
      C
            DO 170  K=1,15
       170  DATA(K) = 0
      C
              M = 3
              NN = 1
              ND = 1
              IF (CLASS .NE. 1)  GO TO 180
              NN = ALL
              ND = MAXPEO
       180    CONTINUE
      C
              DO 250  I = 1, NN, ND
      C
            DATA(M+1) = CLASS
            ITEM = A(I,1)
            IF (ITEM .EQ. 0)  GO TO 240
            TOT = 0.0
       190  NT = STOCKS(ITEM,1)
            FRAC = STOCKS(ITEM,2)/10.0
             TOT  =  TOT + FRAC*P(NT,NR)
            ITEM = STOCKS(ITEM,3)
            IF (ITEM .GT. 0)  GO TO 190
            IF (TOT .EQ. 0.0)  GO TO 210
            A(NN,2) = A(NN,2) + (10*TOT)
            A(NN,3) = A(NN,3) + (100*TOT*TOT)
       210  TOT1 = TOT/100.
            IF (I .EQ. 1)  GO TO 220
            DATA(M+2) = 1000
            M2 = 1000
            IF((PDAM.EQ.1).OR.(PDAM.EQ.3))WRITE(6,7766)CLASS,M2,TOT1
            GO TO 230
       220  DATA(5) = 0
            IF((PDAM.EQ.1).OR.(PDAM.EQ.3))WRITE(6,7766)DATA(4),DATA(5),TOT1
       230  DATA(M+3) = (TOT + 0.5)
       240  IF (A(I,1) .NE. 0)  M = M + 6
       250  CONTINUE
            IF (M .EQ. 3)  RETURN
            IF (PDAM .GT. 1)  WRITE(6,7777) CARDTY, JBASE, (DATA(K),K=1,12)
            IF (TSAR.EQ.1) WRITE(DISK)    CARDTY,JBASE,(DATA(K),K=1,12)
            IF (PUNCH.EQ.1) WRITE(7,8888) CARDTY,JBASE,(DATA(K),K=1,12)
            RETURN
      C     ***************
              ENTRY IRFPT(MAX,MXNT,NCEQUI,STOCKS,P,TGT,EQUIV,
           X                NCPEOP,PDAM,KTEST )
```

```
C       **************
C
          MAXPEC = NCPEOP + 1
          ALL    = NCPEOP + 2
C
          RETURN
C
   7766 FORMAT(' ','CLASS ',I2,'  TYPE', I5, F7.3)
   7777 FORMAT(' ',I2,I4, 15I7)
   8888 FORMAT(I2,I3,15I5)
        END
```

```
          SUBROUTINE RESTAT(A,MA,N,NTRIAL)
          IMPLICIT INTEGER *2 (A-Z)
          INTEGER *4  NAME, A, MA, N
                                    INTEGER *4  TYPES(10)                 *#*
          REAL     *4  TRIAL, AVG, SD, K
                                    REAL     *4   DATA(10)                *#*
    C
            COMMON / LISTER /  ISAVE                                       *#*
    C
          DIMENSION A(MA,3), NAME(2,7)
    C
          DATA NAME /'PEC','PLE','  A','GE ',' PA','RTS',' AM','MO ',
         X          ' TR','AP ','MAT','ERL','  P','CL '  /
    C
    C
          IF (N .EQ. 1)  WRITE(6,1001)
    C
                                    TALLY = 0                             *#*
    C                                                                     *#*
          TRIAL = NTRIAL
          K = 10.*TRIAL
          IF (N .EQ. 1)  NCPEOP = (MA - 2)/2
    C
          ITEM = A(1,1)
          IF (ITEM .EQ. 0)  GO TO 20
          AVG = A(1,2)/K
          SD = (A(1,3)/100. - TRIAL*AVG*AVG)/(TRIAL - 1.)
          IF (SD .LT. 0)  GO TO 5
          SD = SD**(0.5)
          GO TO 10
        5 SD = -1
       10 WRITE(6,1002) NAME(1,N),NAME(2,N), AVG, SD
       20 FLAG = 0
          DO 40  I = 2,MA
          ITEM = A(I,1)
          IF (ITEM .EQ. 0)  GO TO 40
          TYPE = I - 1
          IF ((N .EQ. 1).AND.(TYPE .GT. NCPEOP))  TYPE = TYPE - NOPEOP +999
          AVG = A(I,2)/K
          SD = (A(I,3)/100. - TRIAL*AVG*AVG)/(TRIAL - 1.)
    C                                                                     *#*
                                 TALLY = TALLY + 1                        *#*
                                 TYPES(TALLY) = TYPE                      *#*
                                  DATA(TALLY) = AVG                       *#*
    C                                                                     *#*
          IF (SD .LT. 0)  GO TO 25
          SD = SD**(0.5)
          GO TO 30
       25 SD = -1
       30 IF (FLAG .EQ. 1)  GO TO 35
          WRITE(6,1003) NAME(1,N),NAME(2,N),TYPE,AVG,SD
          FLAG = 1
          GO TO 40
       35 WRITE(6,1004) TYPE,AVG,SC
       40 CONTINUE
    C
            IF (ISAVE .EQ. 0)  RETURN                                      *#*
            IF(TALLY.GT.0)WRITE(7,1111) N, ((TYPES(I),DATA(I)),I=1,TALLY)  *#*
    C                                                                     *#*
          RETURN
     1001 FORMAT('1','RESOURCE     TYPE     AVERAGE     STD DEV',/,
         X          ' ',' CLASS                LOSSES      LOSSES ',/,
         X          ' ','                              PERCENT ',/ )
     1002 FORMAT('0',1X,2A3,'       ALL  ',2(4X,F6.3) )
     1003 FORMAT('0',1X,2A3,I8,2X, 2(4X,F6.3) )
     1004 FORMAT(' ',10X,I5,2X, 2(4X,F6.3) )
     1111 FORMAT(I3, 2X,10(I2, F5.2) )                                     *#*
          END
```

Appendix D

PROGRAMMERS' NOTES

The TSARINA program, as listed in Appendix C, requires about 250K bytes (or 64K words) of core memory. This requirement is the sum of about 115K bytes for program logic and 135K bytes for data storage. If TSARINA is to be run on a computer system that does not support half-word FORTRAN integers (e.g., INTEGER*2) the space required for data storage will be approximately doubled.

If these core requirements are excessive for some installations, they can be reduced in two different ways. First, the program is readily adaptable to a relatively efficient overlay structure that would reduce the space requirements for the program logic. The structure listed below should not affect operating speed seriously and would cut the core required for program logic from about 115K to about 25K bytes:

```
     INSERT MAIN,GAUSS
OVERLAY ZONE
     INSERT INPUT,STORE,SAVER,JMEMO
OVERLAY ZONE
     INSERT BOMB,TESTHT,HITTGT,TGTDIM,TGTORD,TGTZON
OVERLAY ZONE
     INSERT CBU,EXPHIT
OVERLAY ZONE
     INSERT CHECKR,RUNWAY,PLOTHT
OVERLAY ZONE
     INSERT PRINT,STATIS,DAMAGE,REPORT,RESTAT
```

The space required for data storage may also be reduced dramatically, depending upon the nature of the problem of interest to the user. As can be seen from a careful review of the code and comments in the first half of the MAIN routine (pp. 71, 72), the version listed in Appendix C provides space for a relatively large and complex problem: 500 targets, 50 attacks, 1000 resource packets, and hundreds of different types of resources of various classes. For lesser problems, the dimensions of the appropriate storage arrays can be reduced before the program is compiled. With the few exceptions noted in the MAIN routine all of these changes are confined to COMMON statements.

The same procedure can also be used, of course, to increase the size of the storage arrays, if the user's problem exceeds the bounds of TSARINAs current dimensions. With the exception of the numbers of different types of the different classes of resources, there is no practical limit on the size of the problem that can be treated, other than those imposed by available core and the user's budget.